The International DOI Foundation Washington & Geneva

DOI Discussion paper: From one to many
Version: 1.0
Date: August 2000
Status:
The purpose of this paper is *to elicit comments, criticisms, and constructive dialogue.* The statements in this paper should be considered as subject to future amendment. Further versions may be posted to the DOI web site at *http://www.doi.org* and may include changes of policy or proposed direction in light of comments and criticisms received, which are welcome and should be sent to the authors, n.paskin@doi.org *or* llannom@cnri.reston.va.us . This paper may be reproduced and distributed for the purposes of comment, providing that it is reproduced in total including this statement
Circulation: Public

"From one to many": the next stage in the development of DOI functionality

Norman Paskin (International DOI Foundation) &
Laurence Lannom (Corporation for National Research Initiatives)

1. Background

1.1 Purpose of this document

DOI is an actionable identifier: clicking on a DOI achieves some action. In the initial implementation, this is a single action: clicking on a DOI takes a user to one URL. We refer to this here as "one-to-one". "One-to-many" implies one DOI may be used to achieve many possible actions. This is the aim of the full DOI implementation. The purpose of this paper is to explore the mechanisms and consequences for the move from "one-to-one" to "one-to-many". It is put forward for information and to encourage discussion of the issues raised.

1.2 Supporting documentation

This paper builds on the work of the International DOI Foundation (IDF)[1] in developing and deploying the Digital Object Identifier (DOI) system. Some of the concepts discussed in this paper are described in fuller detail elsewhere, and are only described here only as far as necessary to explain the background to an issue. In particular, refer to:

- The DOI Handbook[2]: the most current, and regularly updated, summary of DOI rules and practice.
- DOI: Current Status and Outlook[3]: a summary of the work carried out by the DOI Foundation in its first year of operation, with detailed expositions as to scope and the role of metadata and future directions planned for the DOI (continued in the present document).
- Digital Object Identifier: implementing a standard digital identifier as the key to effective digital rights management[4]: a paper published in April 2000 with a fuller exposition of some aspects of the DOI, building on the previous paper especially regarding deployment (business models and the role of Registration Agencies) and describing issues arising during development.

1.3 Summary of the DOI system

The DOI is an integrated system made up of a number of interacting components that depend on one another for their value. The four primary components are:

- *Enumeration*: assigning an alphanumeric string to the intellectual property entity that the DOI identifies (specified by the DOI Syntax, a NISO standard[5]).

- *Description*: creating a description ("metadata") of the entity which has been identified with a DOI. The interoperable metadata associated with each DOI is derived from principles of the Indecs framework[6], and its implementations in the EPICS data dictionary and the ONIX subset of EPICS[7], which provide a practical basis for the development of DOI Genres. A Genre is a class of intellectual property entities sharing a common set of attributes.

- *Resolution*: making the identifier "actionable" by providing information about what the DOI should resolve to, and the technology to deliver the services that this can provide to users. In the DOI this is provided by the CNRI Handle System® [8].

- *Policies*: the rules that govern the operation of the system

A fundamental emphasis in the DOI design and development has been to ensure that each of these components is extensible: the enumeration syntax allows the maximum flexibility of assignment using de novo or existing identifier sequences; the descriptive metadata follows an extensible framework; the resolution system is part of a wider overall Digital Object Architecture scheme[9,10]; etc. For the purposes of this discussion we deal especially with the two components of *description* and *resolution*.

The DOI system's development to date is in three parallel tracks:

- The *initial implementation* is that which was available at the time of the DOI's conception and launch in 1997/98, and has been in use since. DOIs are assigned by registrants on a relatively flexible basis allowing considerable freedom in the choice of assignment and use. The majority of use is on an experimental or trial basis; DOIs continue to be assigned and supported on this basis; however the full implementation offering additional functionality will supersede the majority of such uses. The registration of DOIs in the initial implementation is supported by IDF's technical partner, CNRI[11].

- The *full implementation* brings together all the components of the system as now conceived; especially it adds functionality via assigned *metadata* and may make use of *multiple resolution*, and implements additional policies (e.g. concerning scope). Applications using DOIs, either full scale services or trial services, are developed using these additional functionalities (which are necessary for the creation of meaningful services). The majority of the several million DOIs registered in the DOI system are now in this category, which is growing rapidly. It is intended that all public DOI applications should use the full implementation, which will become the basis of all DOI registration in the near future. All DOIs that were earlier registered in the initial implementation will eventually be migrated to either an appropriate full implementation, or remain registered with no associated metadata and therefore limited functionality. The registration of DOIs in the full implementation will be supported via a collaborative network of DOI registration agencies operating under common governance by the IDF. In an interim period, registration under the full implementation is also supported by CNRI. The development of DOI Registration Agencies is the other key aspect of DOI development work at present. This is outside the scope of the present paper but is addressed in the DOI Handbook and in a supporting paper on DOI Deployment[12].

- The third track of DOI development involves monitoring and collaborating with other *standards activities* and related organisations (including W3C, ISO, NISO, WIPO, MPEG, etc.), and aims to ensure that DOI develops in concert with other existing and evolving standards.

2. One-to-one

2.1 Initial implementation constraints and simplifications

The initial implementation of the DOI system was based on a deliberate constraint of the handle resolution system to a single resolution from one DOI to one URL. Hence we refer to this as "one to one" resolution. The effect is a redirection from a single name (DOI) to a single piece of associated state data (a single location on the Web, a URL). This is a deterministic "click the link and go there" model of identifier resolution. The associated state data can be updated in the Handle record if it changes; hence a DOI provides a persistent location. This is a useful function, but not the only functionality envisaged for the DOI System. For the full functionality, this constraint is to be removed.

The early experimental implementations of the DOI took place before issues such as scope had been clearly defined and therefore also contained another significant simplification: metadata was not required to accompany a registered DOI. It is now recognised that structured metadata accompanying each DOI (that is, publicly declared and available from within the DOI system) is necessary for the full functionality of the DOI System. Metadata can be added in the one-to-one model to achieve additional functionality beyond the initial implementation. Some metadata is valuable because it is public (e.g. book publishers make author/title/price/date publicly available: this is a useful metaphor for DOI data). Additional, added-value metadata is not necessarily public (or free), and is not part of the DOI required metadata, but can be built following an interoperable data model.

In the initial implementation, with the DOI as a single-point resolution routing system (from a DOI occurrence to a single URL at a publisher-maintained web-site which is the sole source of the entity), there is little need for interoperable metadata, as any related information about the object so identified can be held within that "proprietary" web site. The need for interoperability arises in the use of DOIs as persistent names (identifiers for content entities) which can be used independent of the source of that content. Metadata is essential to commerce as it must be possible to process transactions via unique identifiers without recourse to physical inspection of the items being traded. Processing of DOIs in further applications (e.g. multiple-resolution schemes) necessitate the existence of some basic metadata in order to provide parameters for an intelligent selection process. The DOI is an identifier which can be used to access seamlessly a variety of entities from unrelated sources, and to access different instances of the same resource at different sources: hence the need for a common vocabulary and data model for DOI metadata. This recognition has led to the requirement that even "one-to-one" implementations of DOI will be supplemented by structured metadata. All DOIs that were earlier registered without metadata declaration will eventually be migrated to either appropriate Genres (adding the requisite metadata), or to a default special DOI Genre, the Zero Genre that has no associated metadata and therefore very limited potential functionality.

2.2 Distinguishing what the DOI identifies and what the DOI resolves to

In the one-to-one model, it is important to distinguish what the DOI identifies from what the DOI resolves to. This distinction becomes more clearly apparent (and essential to understand) in the one-to-many model described below. "Click the link and go there" may lead to the potential fallacy that "since this DOI clicks to A, it identifies A".

A DOI on the Internet can be "resolved", leading the user of a DOI to any piece of data that is Internet-accessible. This does not imply that the DOI will necessarily resolve to the entity that it identifies (or an instance of it) - although that will sometimes be the case. The DOI, though, can be used to identify classes of intellectual property - both abstract "works" and physical "manifestations" - that cannot be directly accessed in a digital file. Even when the DOI does identify a digital file, this may not be the appropriate data for the DOI to resolve to. What the DOI identifies from and what the DOI resolves to may be the same thing, but they will often be very different.

Take as an example a physical book (already identified by an ISBN): this may legitimately be assigned a DOI (possibly incorporating the ISBN); hence a DOI can actionably identify it, but it can never resolve to the physical book - only to something about it (e.g. a catalogue entry, an order form, etc). A second example: if the DOI identifies an abstract work, the resolution destination can never be the thing itself (which is by definition abstract) but must be either a derivation of it (one particular digital manifestation) or something about it (a metadata sheet, a catalogue page, etc). Both of these options (a particular manifestation, or a record about the object) are the case in the initial CrossRef implementation of DOI for reference linking of journal articles; the decision on which is used is made by the publisher member of CrossRef.

Multiple resolution, DOI genres, and policies which we are building into the full implementation DOI system are ways of managing all these relationships and associated data in a structured way.

What the DOI identifies is an intellectual property entity.
What the DOI resolves to is a piece or pieces of state data (the current value of some information associated with the entity and defined by the registrant of the DOI).

3. Moving from "Initial implementation" to "full implementation"

The difference between "the initial implementation" of DOI and "the full implementation" of DOI indicated in the three-track description of development is that the full implementation adds *metadata* and *multiple resolution* as additional tools in the inventory to enable DOI use and applications This addition is not a single step, but a number of steps (at minimum two - one of adding metadata, and one of implementing multiple resolution, and each step may be implemented with varying degrees of sophistication) so the transition to a full implementation is along a gradual scale. The aim of DOI functionality development under the aegis of IDF is to analyse the necessary steps further and provide the tools to implement them fully.

An illustrative example (which we will refer to throughout this paper) is the CrossRef implementation of DOIs for journal article reference linking[13] built following earlier IDF prototypes of such application[14]. Metadata may be added to the one-to-one implementation, without adding multiple resolution: in the CrossRef implementation at present one URL is held in each CrossRef DOI record (as in the initial implementation), though mandatory metadata is collected for each DOI (a change from the initial implementation). Within the step of "adding metadata" there may be varying degrees of sophistication of implementation: at present, CrossRef declares associated metadata simply by ensuring that the resolution destination contains, at minimum, the metadata of a bibliographic citation. This means that the kernel metadata associated with the DOI in the "journal article" Genre is discoverable by a CrossRef user. However this declaration of metadata is currently unsatisfactory for wider use of the DOI and extended applications on two counts:

- The discoverable metadata is not structured to allow automated processing: whilst a user may discover the metadata by inspection, there is no process for a machine to discover and use the DOI metadata without human intervention, and then to take that metadata and pass it on for further processing. We would prefer, and will be working towards, a more standard and machine-interpretable declaration (e.g. a standard for a Metadata pointer as a handle record and a standard way of expressing the genre in XML).

- Some metadata is implicit (by context) not explicit: for example the *primary structural type* of the identified journal article as being a Work (abstraction) rather than a specific published form (manifestation); and the *DOI Genre* indicating that this particular DOI is a member of the Journal Article Genre. Within the CrossRef application , these implicit elements are understood and the limitation is therefore acceptable; but it will be dangerous if the same DOI is then used elsewhere, outside the CrossRef system, where the implicit definitions may be lost. (The *primary structural type* distinction is vital for example in considering different applications of identified articles, such as "citation" versus "purchase"[15])

These limitations are not criticisms of the CrossRef implementation; indeed they reflect the fact that the DOI system has not yet developed such specific tools to offer CrossRef, and is now doing so in conjunction with applications like CrossRef which are revealing practical issues to be worked on. We are working closely with CrossRef and other DOI implementations to define and prioritise the next additional requirements. One of our challenges is to work with DOI application developers to ensure that additional features and requirements meet the needs not only of the specific DOI application, but enable wider future interoperability. Particularly in the early days of DOI deployment, it may not be easy from the perspective of a specific application to see the value of some proposed generic DOI system feature, and promoting this generic aspect of development is a key role for the International DOI Foundation.

4. One-to-many

DOI is an actionable identifier: clicking on a DOI achieves some action. "One-to-one" implies that the action is simple and repeatable. In the initial implementation, the action of clicking on a DOI takes a user to one URL.

The real world complexity of all of the potential relations between the identified entity and various associated attributes and services (the sum of which cannot be captured by a single location URL), and the capabilities of the underlying handle resolution system to encapsulate those complexities, require moving the model from a 'one-to-one' model to a 'one-to-many' model. This has been an aim of the DOI effort from the early days of the International DOI Foundation, and is a major reason for the IDF's role as a development consortium. The issue of creating one-to-many functionality is not specific to DOI, but DOI is the first large scale solution for its practical implementation.

"One-to-many" implies one DOI may be used to achieve many possible actions. To make the action predictable (to choose one or more of the results of resolution and process these results meaningfully) requires some additional information specifying the parameters of the action. The tools available to achieve this specification within the DOI system are, by definition, the components of the DOI: *metadata* and *resolution*.

"One-to-many" relationships are a prerequisite for building any intelligent network of relations between identified entities; such a network is necessary to describe the "Semantic Web" concept of W3C, or any data model which describes complex relationships. One-to-many relationships could of course be created in other systems, which could then make use of one-to-one DOIs as simple persistent resolvable identifiers within a wider scheme. However a more worthwhile goal is to make the one-to-many capability a fundamental part of the DOI system, offering a coherent single system able to support E-Commerce in intellectual property:

- The metadata component of the DOI is based on the principles of interoperability of data in e-commerce systems (indecs) which requires unique identification (of intellectual property, of parties, and of agreements) as a core competence and considers metadata as relationships claimed between identified entities. These relationships will usually be one-to-many.

- One-to-many linkage as a mechanism is already present in the underlying handle technology, and this mechanism can therefore be invoked to express (make actionable) the relationships described in the indecs data models.

The DOI has the potential to be the unifying identifier for intellectual property in such a scheme.

4.1 Examples of one-to-many usage

- A DOI which can resolve to a copy of the identified entity or to a metadata record about the entity;
- A DOI which can resolve to several identical copies of the same entity at different locations;
- A DOI which can resolve to several identical copies of the same entity at different locations, but where each location offers different additional services or rights (e.g. aggregators, archives, etc);
- A DOI which can resolve to a copy of the identified entity on my stored local subscription;
- A DOI which can resolve to any of the above, or to some piece of state data offered by a particular service about DOIs

It is obvious that these different situations present a range of different problems and likely different solutions. For example, introducing multiple resolution alone or standard metadata alone will not be sufficient to implement all these examples of usage in which one of many resolution points is appropriate in a particular case.

4.2 Distinguishing two classes of problem: global and local context

The scenario of one resource being available at several, identical, servers (URLs) is a clear case where it would make sense to offer (to everyone using that DOI) the choice of which of these resolution points to use; hence making that choice an attribute of the DOI at a global level, or in a global context. It is seen at its clearest in a case where the resource is free (so that no access control issues arise), e.g. with D-Lib magazine[16]; which is indeed a model where five mirror server URLs are associated in the DOI Handle record with each D-Lib article DOI; resolving a DOI may return one or all the associated URLs. This is being used for experimental and prototype services by CNRI and IDF[17].

The scenario of a resource being available only if certain criteria are met is qualitatively different. For example, I am a member of an institution which has access rights to a particular resource (e.g. a local stored subscription held on a machine with restricted access rights), which is not available to non-members of the institution. I wish, in an ideal world, the action which results from clicking on a relevant DOI to be resolution to that local cache, not to a globally available copy elsewhere; I also wish that not to be the case for someone accessing the same DOI from outside the institution. The choice of which resolution point does not apply to everyone using that DOI, but only to those meeting some criteria; an attribute of the DOI at a local level, or in a local context. This is what has been discussed under the heading of "the appropriate copy problem"[18] (previously referred to as the Harvard problem, after the example cited as a practical case) and is relevant in real world examples where the choice of actions may be influenced by user rights. Note that even if the "local" environment is very large (e.g. suppose a consortium involving "all US .edu sites") the same principle applies; the DOI choice is not a globally applicable one.

The criteria determining the appropriate resolution destination relate to context: entity, user and rights. To automate the process, these criteria must be available in an automatable description. This is the aim of the indecs work; this is why the DOI metadata structure follows the indecs metadata principles and is mapped to an indecs-compliant data dictionary. A suggestion is that if the criteria are independent of user and rights (determined only by the "entity") then this choice could be carried in the DOI Handle record (a "global application"); if the criteria involve user and/or rights information then this choice should not be carried in the DOI Handle record ("local application"), but dealt with by the interaction of the metadata about the entity (discoverable from the DOI) with metadata about the specific context (user, rights). By ensuring that DOI metadata follows a consistent model, such interoperation can be achieved.

4.3 Global context applications via multiple resolution

The ability to associate multiple attributes and/or locations with a single DOI (multiple resolution) has been present in the Handle System since the start of the DOI. In the handle system this is achieved through the association of multiple type/value pairs with each handle[19]. The fundamentals for multiple resolution in the handle consist of two logical components: (1) the data model includes multiple values for a given handle and those values are typed, and (2) the protocol allows for resolution by handle and, within that, by type. So a single handle could resolve to a single URL (the case in the initial DOI implementation), or to multiple URLs, or to multiple values of different types (e.g., two URLs, one e-mail address, and one public key). This typing system is available now and easy to implement.

4.3.1. Developing tools to interpret multiple resolution values

A major challenge to doing this multiple resolution well is that the most commonly used client software, a standard web browser, cannot do anything sensible with multiple returns from a single link, apart from displaying the results and asking the user to make sense of them.

There are two approaches to solving this problem of client-side use of multiple resolution information. The first considers that since this is a problem common to any approach which returns multiple results from a link, and since multiple links are logically necessary to build semantic networks, it seems to be inevitable that tools will be created which overcome this problem, and we could await the development of these tools. XML-capable browsers promise to incorporate this type of functionality and other new clients, such as e-book readers, are now beginning to arrive. These could include such functionality (as well as, ideally, the ability to recognise syntax such as DOI: directly rather than expressed through http proxies, such as already available through use of the Handle plug-in).

The second, and more immediately sensible, solution is to put the use of multiple resolution information into specific lower level applications, running on the current client browsers: e.g. using a metadata pointer in the handle record to gather information about the identified object in order to create some information about the nature of the link (e.g. through a right click on the link which would pop up the various options of data types).[20]

Note that in the discussion of tools development, dealing with multiple returned values is a separate problem from simple syntactic handling of names like DOIs (i.e. ensuring that the browser can deal with *DOI:10.1000/182* as a native syntax keyed into the browser address window, rather than having to use a proxy server and http protocol like *http://dx.doi.org/10/1000/182*) . The two are related, and the HDL resolution protocol has a corresponding API available as a library of C functions. This library, also known as the *client library*, has been used by CNRI in the creation of an http-to-handle proxy server, handle-aware extensions to the Netscape and Microsoft web browsers, a handle caching server, and in various prototype projects. DOI services have been proposed and implemented in prototypes[21] as a way of relating specific meaningful services about a DOI, expressed in a syntax such as *service1@DOI:10.1000/182*, to data type definitions which embody the required resolution result. A syntax has been defined for specifying the resolution request by type; and two clients (a *http proxy* and a web browser extension called the *CNRI Handle System Resolver*[22] often called "plug-in") built that can read that syntax. The ability to make the request by type is already in the Handle client library and works with both http proxy and browser extension clients.

4.3.2 Developing a logical scheme of multiple resolution types

A second challenge is that while it is simple to create many different type codes, and therefore tempting to develop these in an ad hoc fashion in response to each need as it arises, it is not as easy to anticipate which will be useful and which will simply get in the way of the others and confuse users and developers: imagine doing the work of creating and recording this data across millions of identifiers and then finding out that the scheme was wrong and had to change. The syntactic layers (engineering, architecture and data modelling) of Handle resolution ensure that it is all very flexible, which is good, but also means that there are few constraints or standard practices to govern what can be done with it; but those will be required to build anything coherent.

Coherent typing schemes for handle values will need to vary by class of object being identified. For example, a service such as "get pdf" makes sense for a digital object text manifestation, but not for an abstract work; a request to "play" makes sense for a recording but not a still picture. This concept fits well with the concept of DOI genres: the DOI for an entity in a given genre could be expected to resolve to a defined set of typed values and that defined set would vary by genre. A genre would therefore embody not only a set of metadata elements, but also a set of rules about what content typing schemes are meaningful for that genre. This allows the incorporation of some business rules into the genre definition. In the simplest case, the minimal handle value requirements would be a single URL (in some earlier DOI discussions this was referred to as a "level 1 DOI"), while a more complex genre may require one or more content URLs, one or more descriptive metadata pointers, and so on. In such an approach it would seem reasonable to also include a value identifying the genre, and for genres to be registered in a publicly available registry to enable applications to be developed using them.

A logical framework for the semantics of typing is necessary, but is not yet developed. Indeed it may be some way off, but there are indications of how it could be achieved. The indecs work on metadata offers the promise of further elaboration based on the centrality of "events". The concept of events is already central to indecs and other advanced metadata schemas[23] but the details for intellectual property transactions have not yet been fully defined since this requires a deeper description of rights and creation events as well as parties (creators, users, intermediaries, etc) (work on this is planned). The fundamental "model of making" in indecs will then describe events by means of verbs in precise event statements. Some of these verbs might form a logical basis for a set of corresponding actions expressed through data types in resolution. If this could be achieved, it would result in an interoperable set of verbs and attributes, firmly uniting the metadata and resolution components of the DOI. But so far, the semantics of typing multiple values are merely a suggested list of attributes that might turn into types (metadata, sample, buy, rights, pdf, etc) and it is easy to create a long and diverse list, which at present are not readily susceptible to logical typing.

4.4 Local context applications

The second path towards "one-to-many", in addition to global considerations using multiple resolution, has been the development of supplementary or alternative resolution systems which understand the DOI and which can add value to the basic resolution information contained in the core handle system record. The simple notion here is just that the DOI can be used outside of the core resolution system; the next step in possible functionality is that a DOI is used outside the core resolution system but making use of some of the associated DOI metadata.

This second path has been developing primarily in the context of localization (e.g., I may have access to a local copy of a resource which is not, and should not, be recorded in the central resolution system); however "local" does not necessarily imply some geographical restriction, but is more generally an organisational subset of the possible DOI choices (a physically local store is simply one example). So, for example, there is nothing that would prevent an institution from creating lists of DOIs for material it owns and providing its own resolution of those DOIs for its own users and client software. This may or may not involve sending the DOI to the IDF-maintained resolution system. Local systems could stand completely apart or they could supplement and add value to the publicly available resolution information with their own locally relevant information.

A recent collaborative prototype involving CrossRef, SFX, CNRI, and the IDF[24] demonstrated local resolution not by trapping a local request but by associating a redirection (to a local resolver) instruction with the basic resolution request sent to the DOI proxy. The local resolver may then utilize local knowledge, with the default values in the global DOI system, to create a customized and locally meaningful resolution of the identifier. This is a promising approach, but a more extensive prototype is needed to work out a number of issues.

4.5 Parameter passing

A related issue in dealing with DOIs in the context of a specific implementation (the WWW) is the issue of parameter passing. That is, a resolution via the DOI system whereby some additional information is known to the client submitting the DOI resolution, and accompanies the result of a resolution step (i.e. is passed on by the client along with the result of the DOI resolution), thereby adding some useful information.

It is worth stressing that parameter passing (at least as so far encountered in DOI development) is notably different to the issues discussed above in that it seems quite specific to the current world of one specific implementation of the internet (the web and more generically hypertext). The general notion of the DOI, and the associated notions of structured metadata and multiple pieces of typed current state data, will be relevant regardless of the application layer: extensibility and interoperability are key to the design as is the abstraction of fundamental concepts (resolution, description, identification) away from specific implementation layers. Parameter passing is an implementation detail and might look very different in some parallel internet space. Having said that, the current hypertext web implementation is what we have to deal with at present and hence the issue merits consideration.

As an example, consider two links within CrossRef:

- link from Journal A to Journal X
- link from Journal B to Journal X.

In the simplest case (no localization), the two clients use DOI resolution to find Journal X, but then in sending the request to Journal X, the first client sends along a little piece of data that says "Journal A sent me" and the second client sends along a little piece of data saying that "Journal B sent me". So, in theory, the vendors of X know why they got each request. There are some obvious practical applications for this type of information.

Specifying type in a handle resolution query is relatively easy to do using the syntax *(attribute=value)@hdl.* This now works for selected type arguments, e.g. *http://dx.doi.org/(type=URL)@10.1000/53* or e.g. *(type=email)@hdl*; and a naive approach to parameter passing could be to just adopt the same syntax and say that whatever was between the brackets should be kept by the client and then appended to the resolution answer that came back from the handle server: *{x=y}@handle*. However, generalising this example to other possible cases of DOI use shows that this is not so straightforward, because of the potential variability of what could come back. For example, multiple resolution with entries such as public keys as handle values: whilst passing a parameter along with a URL redirect has a clear function, passing a parameter on a public key has no obvious agreed function and meaning at present. To deal with this, it is proposed to make parameter passing an attribute on the type argument, which can be done for URLs and where it might be possible to make some assumptions about defaults, (e.g. assume that unless otherwise specified, what comes back is a URL). An example would then be *http://dx.doi.org/(type=URL{source=JournalofXYZ})@10.123/456*

which, given the proposed default, could be reduced to
http://dx.doi.org/{source=JournalofXYZ}@10.123/456
which would turn into a redirect such as
http://acme.pub/BigScience/Vol27/Iss1/page3.html?DOIparam=source=JournalofXYZ.

This is also an area where some experimentation and further analysis is taking place in the Handle developer and DOI application communities.

5. Conclusions and proposal

The "one DOI to one URL" model is evolving to "one DOI to many different things", via a number of routes which may be used in different circumstances:

- one DOI to multiple URLs or other attributes at the central resolution system;
- one DOI resolvable at multiple locations for added value;
- one DOI with the ability to pass parameter definitions.

In each case, the availability of DOI metadata offers the possibility of automated interaction between the identified entity and some information about context (user, rights, etc.) A one-to-many model allows the creation of semantic intelligence using the DOI system, mapping to the relationships which occur between intellectual property entities in transactions.

DOI applications are suggesting specific problems and possible solutions for the creation of such semantic intelligence in the DOI system; IDF and CNRI developers are considering the generic and interoperation requirements of these developments. Over the next few months, IDF sees this area as one of the two crucial areas of DOI development; the other key area is the commercial deployment of DOIs via Registration Agencies. The issues are definable and susceptible to detailed analysis; the work needed to fully implement everything necessary to put each in place will be challenging but worthwhile.

The most obvious immediate step which can be taken is to ensure the declaration of DOI metadata in a consistent and logical way, since that metadata is the key to all subsequent manipulations of the DOI, whether in global or local applications. That DOI kernel metadata will be required for any operation which needs to compute an algorithm about context - a function of (user+entity+rights). This is a significant step: the first move to multiple data type availability ("multiple resolution") in the DOI system.

As discussed in section 4.3, making metadata available via a data type has two aspects: the logical design, and the tools to implement the design.

- The logical design is to define a data type which is reserved for a pointer to DOI kernel metadata; the corresponding service of the form *metadata@DOI:10.1000/182*; and a standard expression of that declared metadata (XML) for the resolution record; and develop or promote tools able to use this service.
- Design of further data types (more complex services) will be investigated but implemented with caution and initially on an experimental basis

- The tools available to implement this *metadata* service will include the *(type=xyz)@hdl* syntax in both the http proxy and CNRI Handle System Resolver browser extension clients, already available. Services to users can be developed using the http proxy, but only within the usual constraints of what can be done in web browsers.
- Other tools will be investigated: more complex services can be built in middle level tiers: this is what the current larger SFX/DLF prototype we are collaborating in will do. Finally, different kinds of extensions or plug-ins could be built to do different kinds of things, e.g. the "right click" functionality (see 4.3.1) also requires the browser behaviour to be altered and this can be built as an add-in (e.g. Java).

We also intend to continue the current experiments with global typing, with localisation and with parameter passing, and additional experimentation will be encouraged. This will be both in the context of specific application requirements in a practical implementation (e.g. CrossRef), and as a generic development effort of IDF with the aim of providing common tools.

References

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

[2] The DOI Handbook (available as html on the DOI web site, and also available in a pdf version for download): DOI:10.1000/182 (http://dx.doi.org/10.1000/182)

[3] DOI: Current Status and Outlook: Norman Paskin; D-Lib magazine, May 1999.
DOI: 10.1045/may99-paskin (http://www.dlib.org/dlib/may99/05paskin.html)
(Also published in the DOI Annual Review, July 1999)

[4] Digital Object Identifier: implementing a standard digital identifier as the key to effective digital rights management: Norman Paskin, Director, The International DOI Foundation, April 2000 (available as pdf version for download) DOI:10.1000/174 (http://dx.doi.org/10.1000/174)

[5] American National Standard ANSI/NISO Z39.84-2000: Syntax for the Digital Object Identifier.

[6] http://www.indecs.org

[7] http://www.editeur.org

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

[9] Managing Access to Digital Information. Cross-Industry Working Team (XIWT) (1997). http://www.xiwt.org/documents/ManagAccess.html

[10] A Framework for Distributed Digital Object Services; Robert Kahn & Robert Wilensky (1995) http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[11] Corporation for National Research Initiatives; http://www.cnri.reston.va.us

[12] DOI Discussion paper: DOI deployment (Version 2.0 February 2000) available from DOI web site at http://www.doi.org . Note that this paper outlines basic concepts and is now supplemented by detailed information on registration in the DOI Handbook. Refer to IDF for details of current status of registration agencies and deployment.

[13] http://www.crossref.org

[14] Reference Linking with DOIs: A Case Study. H. Atkins et al: D-Lib magazine February 2000. DOI:10.1045/february2000-risher (http://www.dlib.org/dlib/february00/02risher.html)

[15] E-Citations: actionable identifiers and scholarly referencing. Norman Paskin. Learned Publishing vol 13, July 2000 (no.3) pp 159-168. (http://www.doi.org/citations.pdf)

[16] http://www.dlib.org

[17] A demonstration of this multiple URL capability may be seen at http://dx.doi.org/10.1000/176

[18] Choosing the Appropriate Copy: Report of a discussion of options for selecting among multiple copies of an electronic journal article; Digital Library Federation Architecture Committee.

Priscilla Caplan, Dale Flecker, September, 1999; http://www.niso.org/DLFarch.html

[19] http://www.handle.net/introduction.html

[20] A demonstration of this functionality was made at the IDF members meeting (July 2000) by Eamonn Neylon of Informata (eneylon@informata.co.uk)

[21] The Role of Metadata Supply Chains in DOI-Based Value-added Services John Erickson; ICSTI Forum 30 1999. http://www.icsti.org/icsti/forum/fo9904.html#erickson

[22] http://www.handle.net/resolver/index.html

[23] Accommodating Simplicity and Complexity in Metadata: Lessons from the Dublic Core Experience. Carl Lagoze (http://www.cs.cornell.edu/lagoze/Papers/dc.pdf).

[24] Demo page of the CrossRef /DOI/OpenURL/SFX experiment, http://sfxserv.rug.ac.be:8888/public/xref