DDI URN Resolution

Description and Discussion

Draft, 2020-10-13, Joachim Wackerow

Introduction

The DDI specifications use the DDI URN (Uniform Resource Name) as persistent identifier. It is used for any resource which is defined by DDI specifications and can be identified by a DDI URN.

The DDI URN is an URN defined according to the RFC 8141. “Request for Comments (RFC) is a publication from the Internet Society (ISOC) and its associated bodies, most prominently the Internet Engineering Task Force (IETF), the principal technical development and standards-setting bodies for the Internet.” (Wikipedia). “A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that is assigned under the "urn" URI scheme and a particular URN namespace, with the intent that the URN will be a persistent, location-independent resource identifier.” (RFC 8141).

The Namespace Specific String (NSS) of all URNs using the "ddi" NID is a globally unique identifier consisting of the DDI agency identifier (registration authority identifier), the identifier of the DDI resource (data identifier), and the version of the resource (version identifier). This structure is according to the International Registration Data Identifier (IRDI) defined in ISO/IEC 11179 Information technology - Metadata registries (MDR) - Part 6: Registration, Annex A.

The DDI URN is defined in a specific RFC (22 pages) with the title “A Uniform Resource Name (URN) Namespace for the Data Documentation Initiative (DDI)” written by Joachim Wackerow. This RFC will be soon submitted for comments and approval to the related IETF email list (urn@ietf.org). The goal is to achieve a formally approved namespace in the official IANA registry of URN namespaces.

In a subsequent step, the registration for "DDI" in the "URN.ARPA" zone is planned (“ARPA” is an Internet top-level domain). This will enable DNS-based (Domain Name System) resolution of a DDI URN to specific DDI services. A DDI service could enable the resolution of a DDI URN to a physical location like an URL (Uniform Resource Locator).

It should be noted that there are some dependencies on the IETF authorities regarding the steps described in the two previous paragraphs.

This paper at hand describes the multiple levels and steps of the resolution process and the related body of responsibility. The paper points out some issues in the overall setup and raises some related questions.

There are surely more open issues and questions. The document is open for comments.

Resolution Steps

The different steps of the resolution process are listed here:

1. A program (client) sends a request for a specific DDI URN to the DNS (server) with the DNS zone ddi.urn.arpa. The DDI URN is “urn:ddi:us.mpc:PISA-QS.QI-2:1”.
2. DNS delegates the request to the nameserver of the DDI Alliance because this will be the default route for DDI URNs.
3. The nameserver of the DDI Alliance delegates the request to the nameserver of the MPC. The DDI agency “us.mpc” has an entry in the DDI agency registry which defines the MPC nameserver as default route for DDI URNs of the DDI agency us.mpc.
4. The MPC nameserver answers with a list of available DDI services. One of the possible DDI services should be the resolution of a DDI URN to an URL.
5. The program selects one of the DDI services. The program sends a request for the DDI URN “urn:ddi:us.mpc:PISA-QS.QI-2:1” to this specific service.
6. The DDI service sends an answer. This is an URL in the case of the service which can resolve to an URL.
7. The program can use this URL to request the resource which is identified by the DDI URN.

Steps 1 and 5

The program needs capabilities to interact with the DNS, i.e. sending specific requests and receiving list of services.

Responsibility

The application programmer is responsible for this respectively the organization which is interested in DDI software for the use of distributed DDI resources. The programmer can use available program libraries for these purposes.

Open Issue

There are specific program libraries of Colectica available at the DDI Alliance registry site. It needs to be explored whether they are sufficient for the simple use in DDI software.

Step 1 and 2

Prerequisites for the steps one and two are …

- the approved RFC for the DDI URN, i.e. a formal URN namespace for DDI resources,
- the entry “ddi.urn.arpa” in the DNS system, and
- the delegation of DNS requests to the DDI Alliance nameserver.

Responsibility

The DDI Alliance is responsible for this. Joachim Wackerow is working on this.

Step 3

The nameserver of the DDI Alliance needs to be configured in a way that DNS requests are delegated to the specific registered nameservers (in the DDI agency registry).

Responsibility

The DDI Alliance is responsible for this.

Open Issue

It needs to be determined who the actual work is doing. It should be noted that there is some dependency of the University Michigan IT regarding nameservers. One option for doing the work would be Colectica. Colectica is already maintaining the DDI Alliance registry on behalf of the DDI Alliance.

Step 4

The available DDI services of a DDI agency need to be configured in the agency nameserver.

Responsibility

The DDI agency is responsible for this.

Open Issue

DDI services – at minimum the resolution of DDI URNs to physical locations like URLs - are crucial for the distributed use of DDI resources.

Are enough resources and competence available in every DDI agency to maintain DDI services and to configure the agency nameserver accordingly?

Could offers of the DDI Alliance help here? Like support (for example tutorials) for maintaining DDI services at DDI agencies? Or are central DDI services at the DDI Alliance the solution? This would raise other questions like:

- What is the policy of the DDI Alliance on this? This would need to be developed first.
- What are the required resources for this? Can the DDI Alliance afford this?

Step 5

The program needs capabilities to communicate with DDI services.

Responsibility

The application programmer is responsible for this respectively the organization which is interested in DDI software for the use of distributed DDI resources.

Open Issue

DDI services are currently not standardized. Standardized DDI services and related program libraries would enable easier development of software for the use of distributed DDI resources.

A REST-based API for the resolution of DDI URN to URLs and related program libraries seem to be most important.

DDI Vocabularies

DDI Vocabularies are also DDI resources which are identified by DDI URNs. The resolution of these DDI URNs to URL is important for any use of the DDI Vocabularies.

Open Issue

It should be clarified how this can be achieved. There are DDI Vocabularies from the DDI Alliance and from CESSDA. Could a collaborative approach be possible?

Conclusion

The described scenario needs some efforts to provide a workable basis. This will need some time. The “Open Issue” and dependency sections describe some possible obstacles.

Workaround

Until completion of all steps described above, a limited workaround for the resolution of DDI URNs to URLs would make sense. One option is to provide a DDI HTTP-based service which takes the DDI URN as a HTTP query parameter and answers with the related URL. This approach would not require the steps 1 to 4. The investment in the development of this service would not just be for the workaround. This service could be improved later for the final solution.

One option for maintaining the resolution table from DDI URN to URL is the use of the standardized XML Catalog approach. There is related software available – for example in the Java runtime environment. The prototype DDI4R uses this approach.

A workaround approach would raise again the resource questions mentioned above.