

Software source code ID WG

A motivational introduction

Roberto Di Cosmo

roberto@dicosmo.org

December 5th, 2018



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

How we got in touch with RDA

6th RDA CNAM, Paris, 09/15

- first contact
- very few CS people, if any!

9th RDA Barcelona, 04/17

- BoF on Software Preservation ...
- 60+ people in the room!

10th RDA Montreal, 09/17

- Creation of the
Software Source Code IG

11th RDA Berlin, 03/18

- Work on metadata
- Discussion of EU copyright reform

And now...

The *Software Source Code Identification Working Group* (joint RDA/Force 11 effort)



1 Looking for the right identifiers

2 Conclusion

A system of identifiers is

- a set of labels (the identifiers)
- mechanisms to perform :

<i>Generation (minting)</i>	create a new label
<i>Assignment</i>	associate label to object
<i>Retrieval</i>	get object from a label

- optionally, mechanisms to perform:

<i>Verification</i>	check label and object
<i>Reverse Lookup</i>	get label from an object
<i>Description</i>	get metadata of an object

Mechanisms offered in some systems of identifiers



Mech. / System	Handle	DOI	Ark	PURL
Generation	Yes	Yes	Yes	Yes
Assignment	Yes	Yes	Yes	Yes
Retrieval	Yes	Yes	Yes	Yes
Verification	N.A.	N.A.	N.A.	N.A.
Reverse Lookup	N.A.	N.A.	N.A.	N.A.
Description	Yes	Yes	Yes	N.A.

Our challenges in the PID landscape

Typical properties of systems of identifiers

uniqueness, non ambiguity, persistence, abstraction (opacity)

Key needed properties from our use cases

gratis identifiers are free (billions of objects)

integrity the associated object cannot be changed (sw dev, *reproducibility*)

no middle man no central authority is needed (sw dev, *reproducibility*)

we could not find systems with both **integrity** and **no middle man** !

An important distinction: DIOs vs. IDOs

The term “Digital Object Identifier” is construed as “digital identifier of an object,” rather than “identifier of a digital object”

Norman Paskin. 2010

DIO (Digital Identifier of an Object)

digital identifiers for (potentially) **non digital objects**

- epistemic complexity (manifestations, versions, locations, etc.)
- need an authority to ensure persistence and uniqueness

IDO (Identifier of a Digital Object)

digital identifiers (only) for **digital objects**

- can provide both **integrity** and **no middle man**
- broadly used in modern software development (git, etc.)

for the core Software Heritage archive, **IDs are enough**

Limitations of DIOs

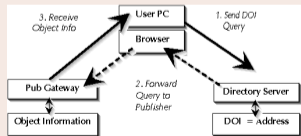
Example: doi:10.1109/MSR.2015.10

- to find what 10.1109/MSR.2015.10 is, go to a *resolver* (e.g. doi.org)
- this returns <http://ieeexplore.ieee.org/document/7180064/>
- at this URL we find ...

The screenshot shows a technical document page with the following details:

- Title:** Mining Component Repositories for Instability Issues
- Page Count:** 1 Page, 45 Total Pages
- Abstract:** Component repositories play an increasingly relevant role in software life cycle management, from software distribution to end-user deployment and upgrade management. Software components shipped via such repositories are equipped with rich metadata that describe their relationship (e.g., dependencies and conflicts) with other components. In this practice paper we show how to use a tool, Distcheck, that uses component metadata to identify all the components in a repository that cannot be installed (e.g., due to unresolvable dependencies), provides detailed information to help developers understanding the source of the problem, and fix it in the repository. We report about detailed analyses of several repositories: the Debian distribution, the Ubuntu package collection, and Cloudfoundry modules. In each case, Distcheck is able to efficiently identify non-installable components and provide valuable explanations of the results. Our experience provides solid ground for generalizing the use of Distcheck to other component repositories.
- Published in:** Mining Software Repositories (MSR), 2015 IEEE/ACM Joint Working Conference on
- Date of Conference:** 01-17 May 2015
- Date Added to IEEE Xplore:** 08 August 2015
- Electronic ISBN:** 978-0-7691-024-2
- IEEE Xplore Number:** 13379103
- DOI:** 10.1109/MSR.2015.10
- Publisher:** IEEE

Architecture of the DOI infrastructure



- DOI resolution *can change*
- content at URL *can change*
- no *intrinsic* way of noticing
- persistence based on *good will* of *multiple parties*



1 Looking for the right identifiers

2 Conclusion

There are many systems of identifiers

- DIOs and IDOs cater to different needs (bit.ly/swhpidpaper)
- IDOs enable **integrity** and **no middle man** properties **together**
 - Software Heritage is using IDOs for billions of objects, **today**
 - we believe IDOs are appropriate for most **digital born** content that has a **canonical** representation

Act before new indicators get standardises (see the OSM!)

- join the RDA SCID WG!
- contribute to understanding how to
 - reference software
 - cite software (not the same!)
- build a consensus