Submitted on: 19.07.2016

Identifiers and Use Case in Scientific Research

Thomas Gillespie
Neurosciences, University of California, San Diego, San Diego, CA, USA
E-mail address: tom.h.gillespie@gmail.com

Qian Zhang
School of Information Sciences, University of Illinois at Urbana-Champaign, USA
E-mail address: zqian1@illinois.edu

Chelakara S. Subramanian
Department of Mech and Aerospace Engineering, Florida Institute of Technology,
Melbourne, FL, USA,
E-mail address: subraman@fit.edu

Yan Han
University Libraries, The University of Arizona, Tucson, AZ, USA.
E-mail address: yhan@email.arizona.edu

Abstract:

The authors have broad representations across of a variety of domains including Oceanography, Aerospace Engineering, Astronomy, Neurosciences, and Libraries. The article is intended to: a) discuss the key concepts related to identifiers and define identifiers; b) explore characteristics of good and bad identifiers in general, along with an overview of popular identifiers; c) demonstrate a use case in a scientific domain of using identifiers via the data lifecycle; c) raise awareness of data collection, data analysis and data sharing for raw and post-processing data for data producers and data users. It is inevitable for multiple identifiers to co-exist in the data lifecycle, as the case study shows different needs of data producers and data users.

Keywords: identifier, use case, data lifecycle

1. Identifiers: Introduction and Definition

A lab's notebook number and page are sufficient for local access in no time. However, sharing and using them by others are not easy. In addition, rows in a spreadsheet and lines on a page both are distinguished, but they are not sufficient for access and stable for future reference, if a new row is inserted and a book has multiple or newer editions. Identifier is critical for providing a stationary point of reference for concrete and conceptual objects. However, identifiers are only useful in so far as they can facilitate aggregation of a community's thoughts, studies, and publications. The members of the community must be the ones to adjudicate the meaning and criteria for meaningful association and/or correctness.

An identifier is a name that we give to an object that we can distinguish it from the rest. It exists in different forms such as name, scientific name, and a string of characters. It can be in any language carrying either semantics or opaque meanings. The concept of identifier is simple but important, as it plays a critical role in our lives. Identifier facilitates communication, reduces confusion and provides linkage across domains. Identifiers are widely used for identifying physical and digital objects. Physical objects include specimen, reagents, lab equipment, instruments (telescopes, hadron colliders..), scrolls, historical artifacts, and more; while digital objects include articles, texts, code, software, data, workflow information, and research environment details to name only a few.

Whether we realize it or not, anyone who collects and records data uses identifiers. One might even argue that anyone who communicates uses identifiers, although words and sequences of phonemes make notoriously poor identifiers. Here we will review some of the properties that all identifiers have in common. Identifiers, from names to words, to GPS coordinates and URLs have two primary functions: one to distinguish and two to reference. While the complexity and intended scope of various identifiers differ significantly, in general identifiers can either be local or global in their scope. Local identifiers may become globally and are required to be updated when data sharing and uses beyond local and original subject domains.

Oxford English Dictionary defines "an identifier" as "a thing used to identify someone or something", and "Computing: A sequence of characters arbitrarily devised to identify or refer to a set of data, a location in a store, or a point in a program" ("Identifier", 2016). These definitions fail to recognize the fundamental functions of an identifier, and yet some identifiers are NOT "arbitrarily devised", but carefully designed with semantics info built in. The authors define that "**An *Identifier is designed to uniquely identify an object*** (e.g. physical object, digital object, data, program, or a byte stream) ***within a certain scope for two functions: to distinguish and to reference.*** An identifier is a sequence of characters, which consists of number, letter, symbol, and/or any combination of those in any languages***.*** In some cases, an identifier may carry its metadata so that more functions (e.g. multiple resolutions) can be developed.

2. Considerations of Identifiers

The localness of an identifier can be qualified based on additional namespaces or identifiers to make it globally unique. For example, a local identifier "pg1" that references page 1 in a book can become globally unique by adding its ISBN in front of the symbol e.g. "ISBN:pg1". In extreme cases, local identifiers may not even be explicit. Implicit identifier systems use certain convenient features of reality to make their references distinguishable. However, such implicit identifier systems are not robust to certain common operations that are performed on data such as insertion and deletion. As a result, implicit identifier systems fail to satisfy distinguishability, reference requirements, and are not persistent.

A corollary of not worth using an explicit identifier system with the cost of adding/updating an identifier. Considering the exceptionally annoying case of manually numbering a line where there is a need to add a new point between 1 to 2 which requires manually renaming every point from 2 to n. The need to promulgate those changes to anyone who is using the old identifier system.

The emerging trends with open science and linked data require data share and data exchange, as science data move from local to global along with local identifiers. Considering time (e.g. Tuesday 8 AM) can be only meaningful in local scope. Mapping from local Tuesday to the global time requires timezone to qualify the local time (8AM).

Referencability:

- Binding: binding is critical to an identifier as it links to an object and/or its metadata about an existence. An example is that call numbers in the old library card catalog. DOI also has linked metadata meeting a specified schema.
- Referential integrity: this term is mostly used in relational database, but it is a concept that allows identifiers exist in multiple systems so that relationships of these systems remain consistent. In the area of open data, this is critical to connect different information space for data share and use.

Transparent or Opaque:

- Transparent (non-opaque) identifiers that contain recognizable semantic strings tend to be easy to read and to identify transcription errors. However, it is difficult to mint them uniquely, quickly and persistently.
- On the other hand, opaque identifiers such as UUIDs are easy and quick to create in large numbers and especially useful for tracking instances, but they are usually long (to be unique) and not useful in certain situations, for example, where a UUID is entered by hand. Guralnick et al., (2015) brought up one possible solution to this dilemma that is to maintain both human-friendly identifiers (e.g., catalog numbers) and computer-friendly identifiers (LOD, UUID, ARK, etc.) for electronic cross-linking, although such solution does require curation overhead to assure that both are managed for the long-term. Emerging services such as GRBio maps human-friendly Institution and Collection Codes to URIs for biocollections.

3. Characteristics of Good and Bad identifiers

Identifiers are designed for a certain purpose. During the data lifecycle, there are multiple players who have different roles and goals to fulfill their specific needs. A person may have one or more roles during the different phases of the lifecyle. Data producer creates data; appraiser selects data; and users uses and share data. The case study below demonstrates a good example of identifiers in a scientific research. As a result, there may have multiple identifiers existing in the lifecycle. The authors believe the following critical features for good identifiers, and certain features to be avoided.

- Referencing: Each identifier shall identify ONLY one unique referent. However, one referent may have multiple identifiers to meet users' needs.
- Persistence: Each identification system shall maintain persistence over the time so that each identifier has persistence not to change its referent. Starr et al., (2015) calls for "unique identification in a manner that is machine-resolvable on the Web and demonstrates a long-term commitment to persistence."
- Binding with metadata: The authors have stated the importance of binding. Other literature stated that both the object and its metadata so that the identifier supports the provision of a human-readable object in one context and machine-readable metadata (e.g., RDF, JSON) in another context (Guralnick et al., 2015).
- Global uniqueness: Open science and open data provide new opportunities for sharing and using data, identifiers and their reference integrity have to reach beyond local to global information space. Duerr et al., (2011) also discussed the suitability of an identifier scheme in the domain of earth science as the capability of providing unique identifiers, unique locators (URNs), serving as citable locators, and uniquely identifying the scientific contents of data objects under format transformations or content rearrangement.

- Consistency in naming: Each identifier shall allow easy implementation of systematic way to assign new identifiers.
- Resolvability: an identifier shall be resolvable under viable protocols, allowing users to reach either the referred object or metadata or both using HTTP or URL currently. There are P2P protocols used in data sharing. In the future, resolvability may be required to be viable via other protocols.
- Use Common English characters and numbers (i.e. a-z, 0-9): This is not a mandatory requirement. Based on our years of experiences in identifier over social science, humanity and scientific fields, it is highly recommend to use common English characters and numbers (a-z, 0-9) and avoid other characters, symbols, languages to reduce the confusion in terms of look-alike but not the same unicode coding.
- Administrative organization: Each identifier shall have a organization that carries out administration and management of the system or protocol. An identification system cannot be sustained itself without an administrative organization.

Our observation and experience identify the following issues with bad identifiers:

- Inconsistence: Inconsistent identifiers can be easily result in collisions.
- Confusing characters: In some identifier systems, users are limited by the design guidelines of underlying character sets for identifiers. Common rules are alphanumberic (e.g. a-z, 0-9, and '-' or '_'). In some identifier systems such as ORCID, system automatically generates identifiers. In some other cases such as DOI and filename, users have their own freedom to design either partial or complete identifiers. In these cases, users shall be aware that certain characters are alike, but they are different in underlying codes. In English, SPACE (U+0020), TAB (U+0009), and NO-BREAK SPACE(U+00A0) are look-alike, but not the same. Using these characters can result in confusion and errors. They carries different meanings and are coded differently in computers with unicodes . Another example is LINE FEED (U+000A), NEXT LINE (U+0085), and CARRIAGE RETURN (U+000D).
- Uncommon direction of written scripts: Most languages in the world are commonly read in top-to-bottom and left-to-right directions. However, right-to-left script are popular in certain languages such as Arabic, Persian and Hebrew scripts. In addition, top-to-bottom and right-to-left scripts were common ways for Chinese, Japanese and Korean in the past. Although right-to-left text is supported in common software, but it creates confusion when generating identifiers.

4. Types of Identifiers

The purpose of this section is to give readers an overview of popular identifiers, and possibly shed a light into choosing one. Over the years, few are not longer relevant and dated. More details shall be consulted with the identifier website and materials. The "ownership" and "size" are two factors to be considered when choosing existing identifiers, as the authors believe that open standards and commonly-used identifiers have much higher possibilities to be maintained and sustained over time. There are many identifiers co-existing for various fields and sectors, as they serve specific business needs, though some have duplication in coverage.

- **ARK**
  - Definition: Archival Resource Key (ARK) is a Uniform Resource Locator (URL) specifically intended for a long-term persistent identifier. (Kunze 2003).
  - Purpose: ARK is developed by the California Digital Library in 2003. ARK has gained some uses in library fields.
  - Ownership: California Digital Library?. Open
  - Size: unknown
  - Syntax: [http://NMAH/]ark:/NAAN/Name[Qualifier]
  - Example: ark:/13030/tf5p30086k

- **Bibcodes**
    - Definition: The Bibliographic Reference code, also known as “refcode”, is a compact identifier to uniquely specify literature reference in a number of astronomical data systems.
    - Purpose: Used in astronomical data systems such as NASA
    - Ownership: NASA, Proprietary
    - Size: unknown
    - Syntax: 19-character identifiers: YYYYJJJJJVVVVMPPPPA (where YYYY is the four-digit year; JJJJJ is a code indicating where the reference was published; VVVV is the volume number; M indicates the section of the journal where the reference was published; PPPP gives the starting page number, and A is the first letter of the last name of the first author. Periods (.) are used to fill unused fields and to pad fields)
    - Example: 1970ApJ...161L..77K
- **DOI:**
    - Definition: Digital object identifier (DOI) is by far the largest used global identifiers.
    - Purpose: Used for citing electronic resources including data. It is a serial code to uniquely identify digital objects with a format of a Prefix and a Suffix, separated by a forward slash. For example: “doi:10.2458/150”. DOI is an ISO standard and also a registered URI with “info:doi/”. The prefix is pre-assigned by DOI registration agency such as CrossRef. The suffix is determined by the publisher. The assigned DOI is permanently assigned to that content regardless of the owner. The current implementation of DOI is the combination of the handle system and a social organization (the DOI foundation) managing the business of the DOI system.
    - Ownership: International DOI Foundation. Open, transparent and ISO 26324
    - Size: 120 million
    - Syntax: prefix[10.xxxx]/suffix
    - Example: doi:10.2458/azu_afghan_ds350_a35
- **Filename**:
    - Definition: Filename is a name used to uniquely identifier a file in a computer file system.
    - Purpose: Different Operation Systems (OS) and file systems impose different rules on filename lengths and allowed characters. It is the most common identifier used for most users.
    - Ownership: OS or media specific; user-defined.
    - Size: N/A
    - Example: /var/www/home/index.html; digital-access.pdf
- **The Handle System**
    - Definition: the Handle system facilitates the assignment and management of unique global persistent identifiers to locate digital resource over time independent of current or future storage locations. (Corporation of National Research Initiatives 2009a, b)
    - Purpose: an array of characters that uniquely identifies the assi The Handle system is used by thousands of organizations to assign persistent identifiers (also called handles).
    - Ownership: CNRI. Open.
    - Size: see DOI
    - Syntax: hdl:prefix/suffix
    - Example:
- **IGSN**
    - Definition: International Geo Sample Number (IGSN) is 9-digit alphanumeric code that uniquely identifies samples from natural environment and related sampling features such as rock specimens, water samples, sediment cores.

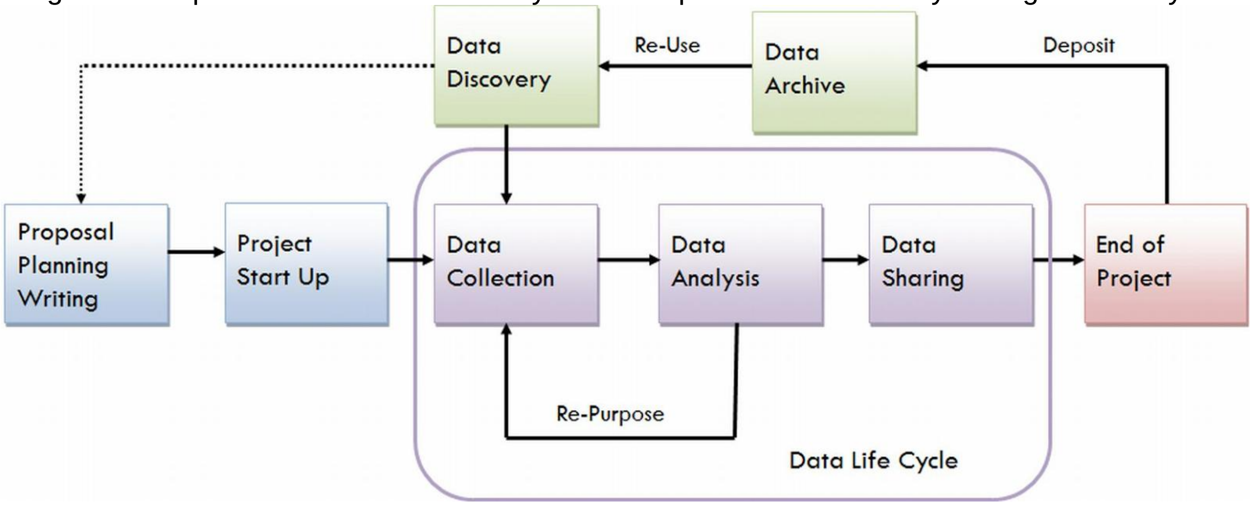
  - Purpose: It is intended to use to uniquely associated with samples. Currently these sample names are ambiguous and often not persistent.
  - Ownership: International Geo Sample Implementation Organization. Open.
  - Size: unknown
  - Syntax: IGSN:9-digit alphanumeric
  - Example: IGSN:HRV003M16
- **LCCN**
  - Definition: The Library of Congress Control Number (LCCN) is a unique prefix and serially assigned number associated with a cataloging record in the Library of Congress in U.S.
  - Purpose: The LCCN has been in use since 1898. Each LCCN number is also. The Library of Congress also provide the LCCN Permalink service, providing a stable URL for all LCCN. info:lccn URI specification:
  - Ownership: Library of Congress. Proprietary.
  - Size: 35 million
  - Syntax: a prefix + year+ serial number
  - Example: 2003556443; gm71002450
- **LSID**
  - Definition: Life Science Identifiers (LSID)
  - Purpose: LSID began in 2003 by the Informatics Infrastructure Consortium (I3C) to uniquely name life science entities. The goal was to "define a simple, common way to access biologically significant data, where that data is stored in files, relational databases, in applications, or in internal or public data sources" (Life Science Identifier Resolution Project, 2010)
  - Ownership: Unknown. Open.
  - Size: unknow
  - Syntax: URN:ISID:<Authority>:<Namespace>:<ObjectID>[:<Version>]
  - Example: urn:lsid:zoobank.org:pub:CDC8D258-8F57-41DC-B560-247E17D3DC8C
- **OCN**:
  - Definition: Online Computer Library Center (OCLC) Control Numbers is a unique, sequentially assigned number associated with a record in WorldCat, a collective collection of the world's libraries.
  - Purpose: OCNs are used to facilitate information exchange from library catalogs to web. They are particularly useful as identifiers for books and other materials without ISBN numbers. The number is included in a WorldCat record when the record is created.
  - Ownership: OCLC, Proprietary
  - Size: 1 billion
  - Syntax: ocm/ocn/on+sequential numbers
  - Example: on9990014350
- **ORCID**:
  - Definition: Open Researcher and Contributor ID (ORCID) is alphanumeric code to uniquely identify authors. It provides a persistent identity for humans, as DOIs for digital objects.
  - Purpose: Used for uniquely identifying an author, not a company, or an organization, or a object. It is particularly important for establishing author metrics.
  - Ownership: Open. non-proprietary
  - Size: 1.85 million
  - Syntax: 16-character identifiers
  - Example: ORCID: 0000-0002-4510-0385
- **PURL:**
  - Definition: A Persistent Uniform Resource Locator. PURL is a URL that is used to redirect to the location of the requested web resource.

    - Purpose: PURLs use HTTP status codes. PURLs have been criticized for their need to resolve a URL to a network location, which has several vulnerabilities such as DNS and host dependencies.
    - Ownership: OCLC. Proprietary.
    - Syntax: see URI/URL/URN section
    - Example: http://purl.oclc.org/OCLC/PURL/FAQ
- **UUID:**
    - Definition: Universally Unique Identifier (UUID) is the cryptographic hash (therefore no collision) of a hexadecimal (128-bit) characters with inserted hyphen characters. The biggest benefit is that it enables distributed systems to identify individual data items or granules without significant central coordination.
    - Purpose: UUIDs are widely adopted in software construction on many computing platforms providing support for generating UUIDs and for parsing their textual information. UUID is also considered as the mostly suitable identifier to uniquely and persistently identify earth science data (Duerr et al., 2011). UUID is one of the examples of identifiers in use for biological samples in the GBIF database (Guralnick et al., 2015).
    - Ownership: the Open Software Foundation; Open, transparent and ISO/IEC 11578 and ISO/IEC 98340-8.
    - Syntax: form 8-4-4-4-12 (totaly 32 lowercase hex characters)
    - Example: 123e4567-e89b-12d3-a456-426655440000
- **URI / URL / URN:**
    - Definition: Uniform Resource Identifier (URI) is a string of characters used to identify a resource. The most common form of URI is the URL, as a web address.
    - Uniform Resource Locator (URL). A URL is a URI which specifies the network location and/or how to access it.
    - Uniform Resource Name (URN): A URI with particular namespace. A URN can be used to locate a resource without implying its action and location.
    - Ownership:
    - Size:
    - Syntax: scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]
    - Examples:
        - URI: ftp://example.org/resource.txt
        - URN: urn:isbn:0451450523
        - URL: https://example.org/absolute/URI/with/absolute/path/to/resource.txt

5. Data Lifecycle and Use Case

There are a few data lifecycle models. One of the most widely used is Digital Curation Center (DCC) data life cycle. The other one that effectively couples data life cycle and research life cycle is from University of Virginia Library, where data management (data collection, data analysis, and data sharing) runs throughout the research project (proposal planning & writing, project start up, and end of project). And the processes in between are not necessarily orderly and linear. It has been observed in most studies (also including the use case in the paper) that some activities in the data life cycle, such as collecting, integrating, and analyzing data, play key roles in the research process of the research life cycle, but the research encompasses more than just the data-centric steps (Carlson, 2014).

Figure 1 Coupled research & data life cycle developed at the University of Virginia Library



So much of identification details (metadata) are provided just to address the needs and levels of understanding of different target audiences seeking the data. We cater three kinds of target audiences: (1) internal group of researchers who are most familiar to our system features and need raw data, (2) outside group who are working in similar research, not so much interested in the raw data but only the physical data, and (3) general public who don't have an in-depth knowledge of our research subject but interested in the general information on hurricanes. A use case can be found at Appendix A to show researchers at the Florida Institute of Technology use a wireless network of sensors to measure pressures, wind speed, wind direction, and temperature on coastal residential buildings rooftop during hurricane or storm events.

6. Summary

The article discussed the key concepts related to identifiers and define identifiers, explored characteristics of good and bad identifiers, overviewed popular identifiers. Finally, the article provided a use case in a scientific domain of using identifiers via the data lifecycle. It is the authors' goal to raise awareness of local and global identifiers used in the data lifecycle through phases such as data collection, data analysis and data sharing for raw and post-processing data. It is inevitable for multiple identifiers to co-exist in the data lifecycle, as data producers and data users play different roles.

Acknowledgement
The authors thank Ruth Duerr for her comments and suggestions.

Reference

Carlson, J. (2014). The use of life cycle Models in Developing and Supporting Data Services. In Joyce M. Ray (Eds.), *Research Data Management: Practical Strategies for Information Professionals*, 63-86. West Lafayette, IN: Purdue University Press.

Duerr, R. E., Downs, R. R., Tilmes, C., Barkstrom, B., Lenhardt, W. C., Glassy, J., Bermudez, L. E., and Slaughter, P. (2011). On the utility of identification schemes for digital earth science data: an assessment and recommendations. *Earth Science Informatics, 4(3),* 139-160.

Guralnick, R.P., Cellinese, N., Deck, J., Pyle, R.L., Kunze, J., Penev, L., Walls, R., Hagedorn, G., Agosti, D., Wieczorek, J. and Catapano, T. (2015). Community next steps for making globally unique identifiers work for biocollections data. ZooKeys, (494), 133.

Identifier. Def. 1a and 1c. (2016). In *Oxford English Dictionary online* (2nd ed.). Retrieved from http://www.oed.com/view/Entry/90998?redirectedFrom=identifier#eid

Kaufmann, R. (2016). *Remote Access Instructions for Hurricane Sensors Deployment, FIT/MAE Internship Report.* Department of Mechanical and Aerospace Engineering, Florida Institute of Technology, Melbourne, FL.

Kunze, J. (2003). Towards electronic persistence using ARK identifiers. In *Proceedings of the 3rd ECDL Workshop on Web Archives.* Retrieved from https://wiki.umiacs.umd.edu/adapt/ images/0/0a/Arkcdl.pdf

Lapilli, G. (2011). *Hurricane Wind Effects: Instrumentation, Measurement and Analysis.* MSAE Thesis. Florida Institute of Technology, Melbourne, FL.

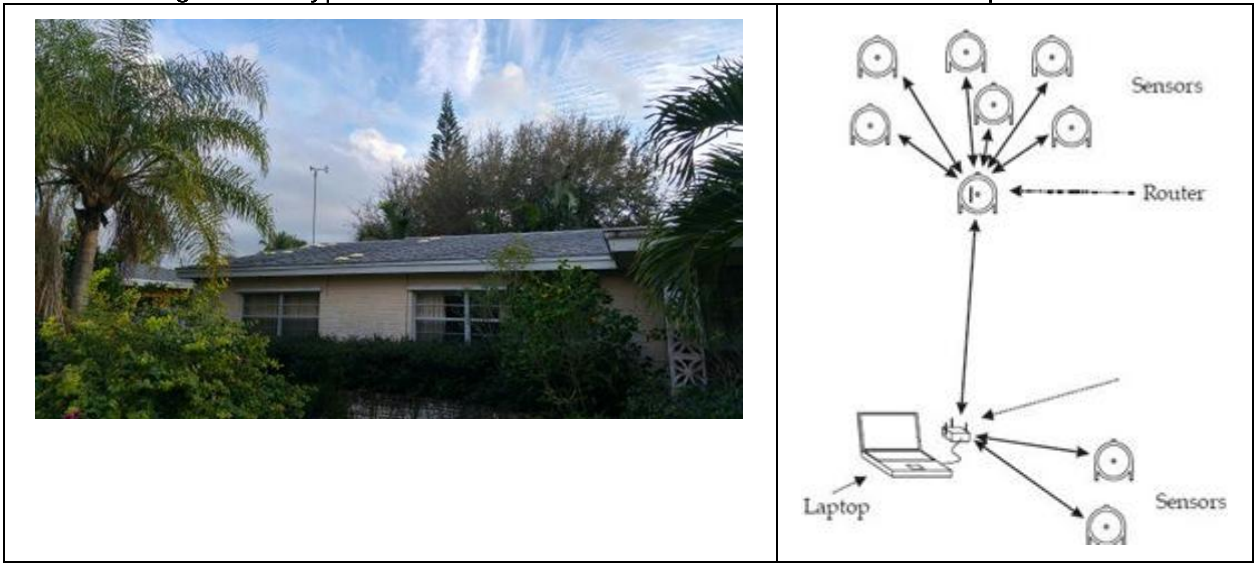
Starr, J., Castro, E., Crosas, M., Dumontier, M., Downs, R. R., Duerr, R., Haak, L.L., Haendel, M., Herman, I., Hodson, S. and Hourclé, J. (2015). Achieving human and machine accessibility of cited data in scholarly publications. *PeerJ Computer Science*, 1, e1.

Appendix A. Use Case: Hurricane and Windstorm Sensing Data

Raw data - Data Collection (initial stage)

There are three systems which can be deployed on three separate houses mounted on roof. Each sensors system consists of up to 30 pressure sensors broken into 3 sub-systems, each with 10 sensors and a router, and one anemometer. (Figure 2) A laptop is used to control and gather the data from sensors, save them and upload them on cloud computing service using Google Drive. (Figure 3) As of now, the raw data and metadata identifiers are purely internal, i.e., no global identifiers have been linked to local identifiers. In the following sections we describe what type of physical data, with what sensors, the format of data and the physical location of sensors, as well as the house GPS coordinates on which the sensors are located. Then metadata programs reformat the raw data, apply sensor calibrations to the raw data, and complete resampling and statistical analyses of the physical variables.

Figure 2: A typical sensor on the roof and sensor data collection process

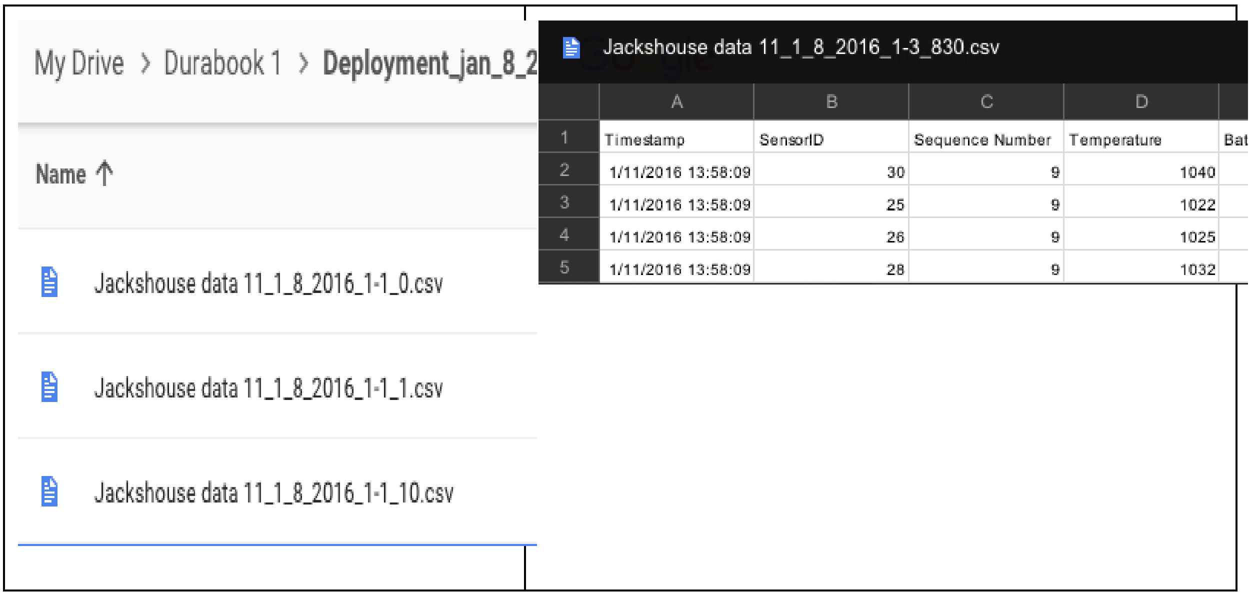


A cloud computing service is used to automatically upload the data saved on the laptop (Kaufmann, 2016). The service also allows researchers to remotely share and use the data. The 3 Durabook folders in Google Drive are dedicated to 3 systems'(3 houses) data. Each folder contains two text documents, configuration files for the sensors system and, a folder entitled "Deployment_month_day_year", named after the particular hurricane deployment and the date. Inside that folder, the raw binary data from all sensors is logged in CSV file <user specified name>_<n>.csv, For e.g., "Jackshouse data 11_1_8_2016_1-1_0.csv (Figure 3). Here n is the sequence number of the 5 minutes data packet. All logging is automatically broken up into separate files each consisting of 5-minute worth of data. The collection of log files associated with a given data collection session are identified as:

<user specified name>_<0>.csv
<user specified name>_<1>.csv
…
<user specified name>_<n>.csv

Where n is the total number of 5 minute chunks logged. The Google drive filenames are local identifiers, and the URLs of shared link of the Google Drive become the global identifier for the raw data share.

Figure 3. Contents of the Google Drive with data files identification



My Drive > Durabook 1 > Deployment_jan_8_2

Name ↑

Jackshouse data 11_1_8_2016_1-1_0.csv

Jackshouse data 11_1_8_2016_1-1_1.csv

Jackshouse data 11_1_8_2016_1-1_10.csv

Jackshouse data 11_1_8_2016_1-3_830.csv

	A	B	C	D	
1	Timestamp	SensorID	Sequence Number	Temperature	Bat
2	1/11/2016 13:58:09	30	9	1040	
3	1/11/2016 13:58:09	25	9	1022	
4	1/11/2016 13:58:09	26	9	1025	
5	1/11/2016 13:58:09	28	9	1032	

Data documentation and metadata file: All rows have the same number of columns, and each column represents the same field. A microsecond-resolution timestamp is prefixed to each log entry indicating the time of the receipt of the data being logged. The following Figure 4 shows the buffer stream segment of a typical output file. Note that: "_" = Blank Space used for separating columns

Figure 4. Segment of a buffer thread of a typical output (raw data) file generated by WINDS-HM data acquisition Program

```
77114.46_T_1_178
77114.56_R_1_543
77114.66_B_1_649
77114.77_P_1_836_-3_5_-1_2_4_0_-2_-2_-1_...
77114.88_T_2_178
77114.97_R_2_543
77115.08_B_2_649
77115.19_P_2_836_-2_4_1_-2_-3_1_-2_-1_-3_...
77115.28_T_3_168
77115.37_R_3_532
77115.48_B_3_549
77115.59_P_3_801_-2_3_1_-1_-2_2_-2_-1_-4_…
```

The description of columns in the data file is as follows.

(1) Timestamp - column 1

The timestamp records to the nearest hundredth of a second with reference to midnight (0:00 hrs). For example: 77114.46 secs means 21:25:14:46 hrs. In other words: 21*(60mins*60secs) + 25*(60secs) + 14 secs + 46/100 secs = 77114.46 secs. After passing midnight, the timestamp will start all over from 0:00 hrs, but the name of the output file will contain the date so data is not overwritten or having multiple files with the same name.

(2) Signals Types - column 2

There are six measured signal types. They are designated by the first letter after the time stamp. The signals are:

- P = Pressure samples (minimum of 4 and maximum of 128 per cycle) - Sensor
- S = Wind speed cycle samples (minimum of 4 and maximum of 128 per cycle) - Anemometer

- T = Temperature sample (only one per cycle if it is commanded) – Sensor and Anemometer
- R = RSSI (Signal quality) sample (only one per cycle if it is commanded) – Sensor and Anemometer.
- B = Battery charge sample (one per cycle if it is commanded) – Sensor and Anemometer.
- D = Wind direction sample (one per cycle and if it is commanded) - Anemometer

(3) Sensor ID Number - column 3

The identification of every remote unit, both anemometers and sensors, is made using integers from 1 to 99 or greater. With the current system there can be 30 sensors and 2 anemometers. The first anemometer is marked with ID number 31, while the second anemometer is marked with ID number 32. The format of this output is a fixed 2 digital number extension.

(4) Digital Unit Sample(s) - column 4

With a 10 bits A/D converter, digital unit samples are integers that range between 0 and 1024. They represent the actual values of the signal (pressure, temperature, RSSI, wind direction, wind speed, or battery) from the A/D converters bits. These data are unformatted and have no calibration. The first integer in the line stream after the remote unit ID corresponds to the real value of the signal, which is referred to as the pivot signal. The data that follows gives steps-change of the signal (pressure/wind speed) in regards to the pivot value (usually a small number like -5 or 2). In order to tell the absolute sample-values from the differentials, a serial algebraic sum needs to be calculated for the pivot sample (first sample). For example, the buffer at 77114.77 seconds may be calculated from the samples in differential digital units as follows:

Buffer*: 836_-3_5_-1_2_4_0_-2_...*

Reconstructing absolute pressures:

[836] [836-3] [836-3+5] [836-3+5-1] [836-3+5-1+2] [836-3+5-1+2+4] [836-3+5-1+2+0] [836-3+5-1+2+0-2] …

Reconstructed absolute pressure buffer: *836 833 838 837 839 843 843 841 …*

A metadata file, prefixed by the homeowner’s name,e.g., “Jackshouse”, contains the coordinates of the pressure sensor locations for every deployment.

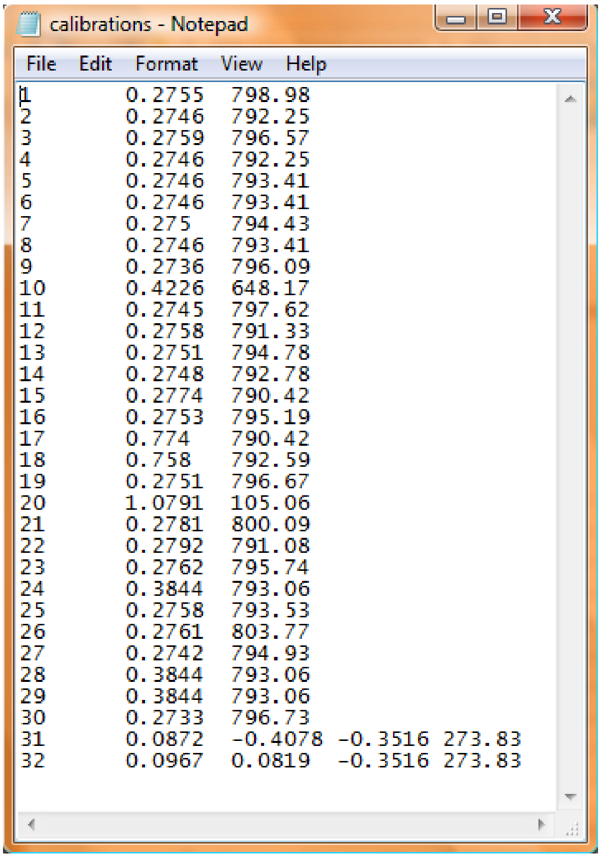
Post-processing Data: Data Collection with Calibration

A post processing code was developed by Lapilli (2011) and Kaufmann (2016) in MATLAB in order to sort and convert all the information collected by the systems into usable data. The program consists of two main sections: File Retrieving and Data Processing

- File Retrieving: The program displays a standard Windows “Open File” dialog. Selecting one of the 5-minute files in the dataset is enough to convert the entire range of data collected by a subsystem. The directory contents are read and sorted, then the header of the first file is read in order to determine the sample rate.
- Data Processing: Files are read in order from the first to the last. Lines are read at the same time and stored in a cell array. Each row is then read individually to identify whether it belongs to an anemometer or pressure sensor. Then a simple algorithm checks if the sequence ID and timestamp correspond (meaning no apparent time drift has happened, compared to the last packet processed for each sensor), while it also aims to discover packets that have been retransmitted and arrived in different order to the base.

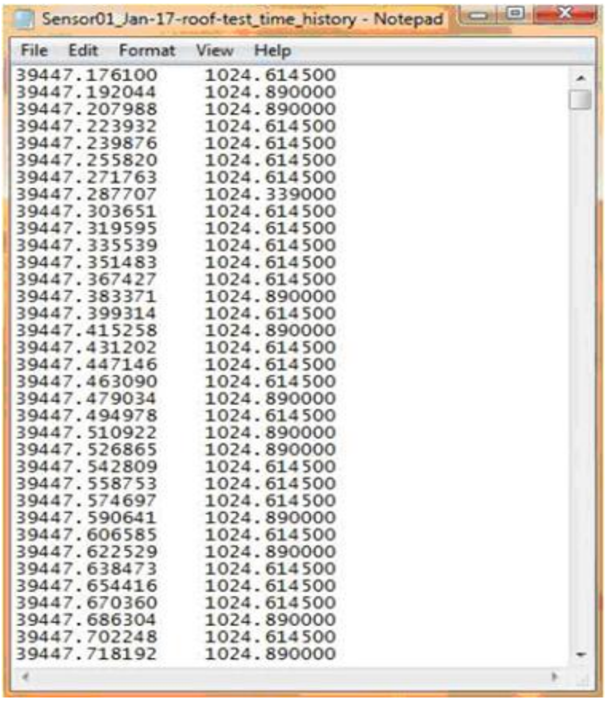
When the post-processing program is started, the calibration curves are automatically read and used in the conversion process (pressure is converted from digital units to millibars, wind speed is converted from digital units to meters per seconds, and wind direction is converted from digital units to degrees and temperature from digital units to degrees Fahrenheit). During the analysis of the raw data, the post-processing program converts the raw measurements into the appropriate physical units. Using the calibration curves, which are recorded in the file “calibration.txt”, the raw data are converted. The calibration file is a simple text file that can be opened and edited using any text editor. Each row in the calibration file (Figure 5) is dedicated to one of the sensors. The first column represents the sensor's assigned number for the deployment. The remaining columns contain the calibration parameters. The number of calibration parameters depends on the sensor type.

Figure 5 Calibration file for the post-processing program



```
1           0.2755   798.98
2           0.2746   792.25
3           0.2759   796.57
4           0.2746   792.25
5           0.2746   793.41
6           0.2746   793.41
7           0.275    794.43
8           0.2746   793.41
9           0.2736   796.09
10          0.4226   648.17
11          0.2745   797.62
12          0.2758   791.33
13          0.2751   794.78
14          0.2748   792.78
15          0.2774   790.42
16          0.2753   795.19
17          0.774    790.42
18          0.758    792.59
19          0.2751   796.67
20          1.0791   105.06
21          0.2781   800.09
22          0.2792   791.08
23          0.2762   795.74
24          0.3844   793.06
25          0.2758   793.53
26          0.2761   803.77
27          0.2742   794.93
28          0.3844   793.06
29          0.3844   793.06
30          0.2733   796.73
31          0.0872   -0.4078 -0.3516 273.83
32          0.0967   0.0819  -0.3516 273.83
```

Figure 6 A time history file produced by post-processing the output (raw data) files


Sensor01_Jan-17-roof-test_time_history - Notepad
File Edit Format View Help

```
39447.176100     1024.614500
39447.192044     1024.890000
39447.207988     1024.890000
39447.223932     1024.614500
39447.239876     1024.614500
39447.255820     1024.614500
39447.271763     1024.614500
39447.287707     1024.339000
39447.303651     1024.614500
39447.319595     1024.614500
39447.335539     1024.614500
39447.351483     1024.614500
39447.367427     1024.614500
39447.383371     1024.890000
39447.399314     1024.614500
39447.415258     1024.890000
39447.431202     1024.614500
39447.447146     1024.614500
39447.463090     1024.614500
39447.479034     1024.890000
39447.494978     1024.614500
39447.510922     1024.890000
39447.526865     1024.890000
39447.542809     1024.614500
39447.558753     1024.614500
39447.574697     1024.614500
39447.590641     1024.890000
39447.606585     1024.614500
39447.622529     1024.890000
39447.638473     1024.614500
39447.654416     1024.614500
39447.670360     1024.614500
39447.686304     1024.890000
39447.702248     1024.614500
39447.718192     1024.890000
```

In the time history file, each row only contains two entries. The first entry is the time in seconds from midnight. (Midnight will be 0, then and 16:20:43 will be 58843). The second entry represents the measured quantity. For example, rows that are shown in Figure 6, report time in seconds and pressure in millibars.In addition to pressure, the post-processing program also creates a complete time history of: 1) the sampling rate, 2) the battery level, and 3) the temperature. Three time-history files are generated for each sensor, with each file containing two columns/data entries. For pressure sensors only two parameters are given. Several parameters must be set before the program can be started. -The operator must enter the name of the output files that will be generated by the post-processing files. The beginning of the file name string and its extension cannot be changed ("Sensor01" and ".txt" need to remain). The rest of the string can be changed at will.

Post-processing Data: Data Analysis with Re-sampling

At this point in the post-processing, the pressure, temperature, wind speed, wind direction, battery level, sampling rate, and the temperature data have been divided and rearranged into different files (one file for each sensor). This allows the different data to be plotted and analyzed. However, when studying a three-day deployment it is not necessary to study all the data recorded at a rate of 25 to 80 Hz. For this reason, the post-processing has a second part, re-sampling. The re-sampling part was designed to average the data collected over a certain amount of time (1s, 3s, 10s, 1 min… up to ½ hour). Averaging the data allows the noise collected during data acquisition to be minimized. In addition to noise, the wireless communication can cause data to be corrupted. In the re-sampling phase, the program goes back over all of the recorded data. The program reviews the data and eliminates the unrealistic ones. All of the data that is outside of the range of the possible realistic will be eliminated to make the time history better.

The second part of the post-processing uses the output data files from the previous part as input files. Like in the first part, there will be one file for each sensor used during the deployment. Next, the program calculates the average over the desired time period and writes that average into files. The format of the time is modified to "month_day_year_hours_minutes_seconds" instead of time in seconds from midnight. In addition to the pressure average, the minimum and maximum pressures over the time period are recorded in these files. After all of the sensors have been re-sampled, the program will also re-sample the wind direction data for the anemometers.

At this point of the post-processing, the pressure, wind velocity and wind direction files have been re-sampled. They are now in the proper format to be used by the plotting program. The main goal of the plotting program is to plot the deployment data. The program will plot all of the pressure data onto the same plot, which will allow for analysis of the differences in pressure recorded by the different pressure sensors at different locations. The program will also make a second plot of the wind speed and wind direction. This plot can later be compared to the pressure plots to look for a correlation between pressure change and wind speed.

The plotting program also has a second purpose. This part is essential when dealing with pressure sensors recording at a so high frequency rate, because it is necessary to smooth the plot and reduce noise due to data acquisition. In order to do this, the spectrum of the sensor's signal was analyzed. After tracing the frequency spectrum of the pressure data for several deployments, and for different sensors, the main signal (the real sensor measured) appeared at the low frequencies of the signal.

Data Sharing

This data is then used by wind engineering research communities for comparison with other full-scale and model simulation results as well as, for further scientific analysis of hurricane characteristics and their impact.