

Generalizable Middleware to Support Use of REDCap Dynamic Data Pull for Integrating Clinical and Research Data

Thomas R. Campion, Jr., PhD^{1,2,3,4}, Evan T. Sholle, MS¹, Marcos A. Davila, Jr., MS¹
¹Information Technologies and Services Department, Weill Cornell Medicine, New York, NY; ²Clinical and Translational Science Center, Weill Cornell Medicine, New York, NY; ³Department of Healthcare Policy and Research, Weill Cornell Medicine, New York, NY; ⁴Department of Pediatrics, Weill Cornell Medicine, New York, NY

Abstract

To support integration of clinical and research data, the makers of REDCap, a widely-used electronic data capture system, released the Dynamic Data Pull (DDP) module. Although DDP is a standard module in REDCap, institutions must develop custom middleware web services to exchange data between REDCap and local source systems. The lack of middleware is a barrier to institutional adoption and use by investigators. To overcome this gap, we developed a REDCap DDP web service middleware (accessible at <https://github.com/wcmc-research-informatics/redcap-ddp>) that minimizes developer effort, relies on configuration by non-developers, and can generalize to other settings. Early findings suggest the approach is successful.

Introduction

Clinical and translational research relies on the successful integration of workflows and computerized information systems that support patient care and research (1). Historically electronic health record (EHR) and electronic data capture (EDC) systems have enabled collection of clinical and research data, respectively, but studies using EDC systems for the completion of electronic case report forms (eCRFs) increasingly require data from EHR applications. Efforts to integrate electronic clinical and research data include periodic extraction of study-relevant clinical data from EHRs (2), embedding of eCRFs in EHRs (3), deployment of research applications that generate clinical documents for storage in EHRs (4), and pre-population of eCRFs with EHR data (5) as part of a comprehensive clinical research tool platform (6). Although innovative methods for integrating clinical and research data through EHR systems exist, adoption of such approaches across academic health centers has remained low over time (7–9).

To support integration of clinical and research data, the makers of REDCap, a widely-used EDC system among non-profit health institutions (10), released the Dynamic Data Pull (DDP) module (11) as part of version 5.9.0 of the software. Using a unique record identifier, such as a medical record number (MRN), DDP queries a remote source, such as an EHR system, to retrieve one-time data elements, such as patient demographics. Use of an optional timestamp parameter can facilitate DDP's retrieval of temporal data, such as laboratory results, for a specific patient. After retrieving data, DDP allows users to adjudicate elements prior to saving results in eCRFs. Investigators at Vanderbilt University Medical Center, where the REDCap consortium is based, use DDP extensively (11).

Although DDP is a standard module in REDCap, institutions must develop custom web services to exchange data between REDCap and local source systems (12). The lack of middleware is a barrier to adoption of REDCap DDP by institutions, use of REDCap DDP by investigators, and integration of clinical and research data across the biomedical research enterprise. To overcome this gap, we developed a REDCap DDP web service middleware that minimizes developer effort, relies on configuration by non-developers, and can generalize to other settings. The purpose of this paper is to describe our middleware approach, demonstrate its use in one institution, and encourage replicability of the approach in other settings to increase use of REDCap DDP for integrating clinical and research workflows.

Methods

As described below, middleware for REDCap DDP consists of a field dictionary, project-specific configuration files, project whitelist, and web services. We designed the middleware to use configuration rather than custom programming.

Field dictionary

The field dictionary is a JSON file containing the complete set of data elements available to REDCap. As depicted in Figure 1, Each dictionary entry consists of a JSON element with six attributes. The "Name" attribute identifies the

dictionary element and is matched to the "Dictionary" element of a configuration file during runtime. "SQL" is a structured query element (SQL) statement that returns a result set given an MRN. One-time fields require an MRN and return only the data requested while temporal fields require a date and return a tuple of the data and the date in the result set. Aliases for data columns are required if the column uses an aggregation function, retrieves a column used in a join, or is dynamically computed. "Column" specifies the data column or alias in the result set while "Anchor Date" specifies the date associated with that data for a temporal field. "Source" is used to identify what server to connect to retrieve the data. Finally, the "Temporal" attribute marks a dictionary term as static or temporal. REDCap DDP currently supports population of free text fields and radio buttons in eCRFs; check boxes are not supported.

```
{
  "mrn": {
    "Name": "mrn",
    "SQL": "select identity_id from
           super_daily.dbo.identity_id
           where identity_type_id = 3",
    "Column": "identity_id",
    "Anchor Date": "",
    "Source": "ARCH",
    "Temporal": false
  }
  "ancillary_testing": {
    "Name": "ancillary_testing",
    "SQL": "select ancillary_testing, DATE_OF_SURGERY from
           super_adhoc.dbo.gic_colorectal_info gci inner join
           super_adhoc.dbo.gic_patient_dem_info gpdi
           on gpdi.dem_id = gci.dem_id
           where NYP_MRN = id and DATE_OF_SURGERY
           between timestamp_min and timestamp_max",
    "Column": "ancillary_testing",
    "Anchor Date": "DATE_OF_SURGERY",
    "Source": "ARCH",
    "Temporal": true
  }
}
```

Figure 1. Dictionary items specify remote source data available to all projects.

Configuration files

As shown in Figure 2, each project has a specific configuration file comprised of entries with ten attributes. Of these, only the "Field", "Label", "Temporal", "Identifier", and "Dictionary" attributes are required. "Field" and "Label" are visible within REDCap and identify each entry. "Temporal" flags a configuration term as either one-time or repeating value, which indicates to REDCap whether dates will be required or not. "Identifier" marks the configuration term that maps to the medical record number and is NULL otherwise. "Dictionary" maps configuration file terms to terms in SUPER REDCap's dictionary, which the web service pairs together based on name. The two attributes of "time_format" and "map" are utilized internally by the data web service for formatting and conversions.

The "time_format" attribute unifies the way REDCap receives dates from a source system. Converting dates within SQL and passing the result set to PHP can invoke undesirable behaviors of the PHP 5 date libraries. One such example of this undesirable behavior is defaulting to the value '1969-12-31' when PHP is unable to parse a date. "time_format" enables the web service to process conversion within PHP, defaulting to "Y-M-D" when no value is supplied.

The "map" attribute translates database column values into the numerical identifier of the REDCap radio button values. If any mapping is required, the left side specifies the source values and the right side the target values. This step is skipped if no mapping is specified. For example, gender can be represented as "M" or "F" in a database. The target values in the REDCap form are "Male" and "Female", and these radio buttons have identifiers of "1" and "2"

respectively. The web service checks the "map" attribute, which would instruct the web service to translate "M" into "1" and "F" into "2" and returns those values to REDCap. In Figure 2, the configuration entry for "ethnic_group" associates remote source system values (e.g. "HISPANIC OR LATINO OR SPANISH ORIGIN," "ASIAN / PACIFIC ISLANDER") with REDCap variable keys for "Hispanic" as defined by "1", "Not Hispanic" as defined by "2," "Declined" as defined by "3," and "Other" as defined by "4." The other attributes are optional and organize terms displayed in the REDCap field mapping screen into categories and subcategories.

```

"1076": {
  "mrn": {
    "field": "mrn",
    "label": "MRN",
    "description": null,
    "temporal": 0,
    "category": "Demographics",
    "subcategory": null,
    "identifier": "1",
    "time_format": "",
    "dictionary": "mrn",
    "map": ""
  }
  "ethnic_group": {
    "field": "ethnic_group",
    "label": "Ethnicity",
    "description": null,
    "temporal": 0,
    "category": "Demographics",
    "subcategory": null,
    "identifier": null,
    "time_format": "",
    "dictionary": "ethnic_group",
    "map": {
      "HISPANIC OR LATINO OR SPANISH ORIGIN": "1",
      "AFRICAN AMERICAN": "2",
      "AMERICAN INDIAN / ESKIMO": "2",
      "ASIAN / PACIFIC ISLANDER": "2",
      "CAUCASIAN": "2",
      "NOT HISPANIC OR LATINO OR SPANISH ORIGIN": "2",
      "DECLINED": "3",
      "NULL": "4",
      "": "4",
      "MULTI-RACIAL": "4"
    }
  }
},

```

Figure 2. Configuration files specify dictionary items and transformations for specific projects denoted by project ID.

Whitelist

A whitelist of REDCap projects permitted to use DDP exists in a file called constants.php. Each project in the whitelist has a configuration file specified along with a REDCap project id. The web services will issue an error message if a project ID provided by REDCap does not exist in the whitelist.

Web services

Ten PHP files comprise the middleware. Each of the three web services—authorization, data, and metadata—has an eponymous PHP file while two additional PHP files serve as unit tests. The REST framework is contained within index.php, which waits for a request from REDCap and inspects the number of arguments provided to determine which web service to invoke. Two additional PHP files, ConfigDAO and FieldDictionary, define classes for data

access objects that store configuration file and dictionary data, respectively, read from configuration and dictionary JSON files.

Figure 3 depicts the flow of data through the middleware. Given a REDCap project ID, the metadata web service instantiates a ConfigDAO object containing dictionary items, time formats, and mapping values as defined in the project-specific configuration JSON file. The metadata service then returns JSON-encoded results to REDCap based on the custom project settings.

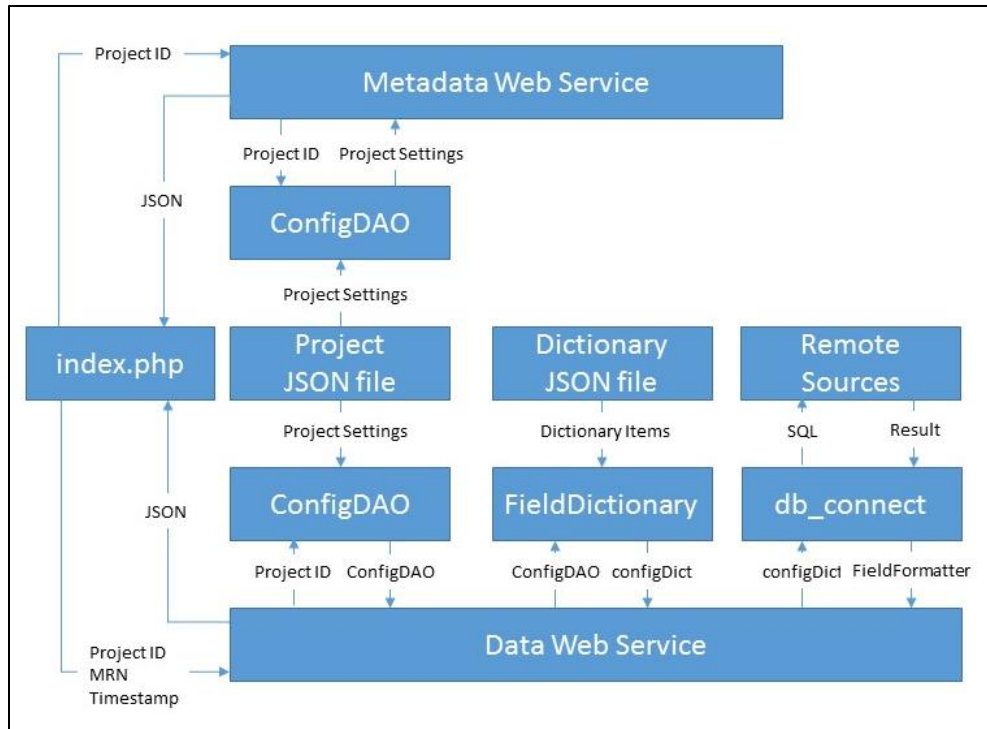


Figure 3. Configuration files specify dictionary items and transformations for specific projects denoted by project ID.

For a specified REDCap project ID, the data web service instantiates a ConfigDAO object followed by a FieldDictionary object containing all dictionary items from the dictionary JSON file, removes dictionary items from the FieldDictionary object that do not pertain to the project as specified in the ConfigDAO object and configuration JSON file, and applies parameters sent from REDCap, such as MRNs and timestamps, to create a “configDict” object containing SQL statements for execution on remote source systems. The data web service can connect to different remote data sources using multiple drivers (e.g. Microsoft SQL Server, Oracle, MySQL) as specified in the dictionary and processed by db_connect.php. After the service retrieves data, code from the REDCapFieldFormatter PHP file applies project-specific settings from the "configDict" object to format and parse the result set. The REDCapFieldFormatter has rules for formatting JSON to return to REDCap for requested one-time and temporal fields. The data web service then returns JSON-encoded results to REDCap for the specified project.

Results

Using the approach described above, we successfully deployed REDCap DDP for Weill Cornell Medicine investigators. To date, five projects have used REDCap DDP. Informatics staff involvement has included business analysts configuring dictionary and project-specific configuration files and developers making minimal code changes.

Source code is available on Github <https://github.com/wcmc-research-informatics/redcap-ddp> through an open source license. At the time of this writing, we have provided source code for the middleware to at least three institutions in response to questions posted on the REDCap listserv about implementing DDP webs services.

Discussion

Although REDCap DDP lacks web services to interact with data sources at local institutions, the middleware described in this paper has successfully supported REDCap DDP use at our institution and may generalize to other settings, potentially reducing the time required for informatics groups to offer DDP to researchers and more quickly streamlining the integration of clinical and research data workflows for investigators.

Limitations of the current study include analyst configuration of JSON files being error prone as well as a lack of a formal evaluation. Future work will replace JSON field dictionary and project-specific configuration files with REDCap projects to enable easier setup of items. Additionally, future work will evaluate the effect of the middleware on informatics and study staff with respect to time required for development, maintenance, and data collection as well as other measures. Furthermore, future study will investigate differences in EHR vendor, clinical terminology, medical specialty, clinical research workflow, and other organizational characteristics across sites implementing the middleware.

REDCap DDP can facilitate the integration of clinical and research data, and the middleware described in this paper may enable institutions to more quickly implement it.

Acknowledgements

This study received support from NewYork-Presbyterian Hospital (NYPH) and Weill Cornell Medical College (WCMC), including the Clinical and Translational Science Center (CTSC) (UL1 TR000457) and Joint Clinical Trials Office (JCTO).

References

1. Payne PRO, Johnson SB, Starren JB, Tilson HH, Dowdy D. Breaking the translational barriers: the value of integrating biomedical informatics and translational research. *J Investig Med* [Internet]. 2005 May [cited 2016 Sep 22];53(4):192–200. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/15974245>
2. D'Avolio L, Ferguson R, Goryachev S, Woods P, Sabin T, O'Neil J, et al. Implementation of the Department of Veterans Affairs' first point-of-care clinical trial. *J Am Med Inform Assoc* [Internet]. 2012 Jun [cited 2014 Jan 14];19(e1):e170-6. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3392864&tool=pmcentrez&rendertype=abstract>
3. Marsolo K, Margolis PA, Forrest CB, Colletti RB, Hutton JJ. A Digital Architecture for a Network-Based Learning Health System: Integrating Chronic Care Management, Quality Improvement, and Research. *EGEMS* (Washington, DC) [Internet]. 2015 [cited 2016 Sep 22];3(1):1168. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/26357665>
4. Kush R, Alschuler L, Ruggeri R, Cassells S, Gupta N, Bain L, et al. Implementing Single Source: the STARBRITE proof-of-concept study. *J Am Med Inf Assoc* [Internet]. 2007/06/30. 2007;14(5):662–73. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/17600107>
5. El Fadly A, Rance B, Lucas N, Mead C, Chatellier G, Lastic P-Y, et al. Integrating clinical research with the Healthcare Enterprise: from the RE-USE project to the EHR4CR platform. *J Biomed Inform* [Internet]. 2011 Dec [cited 2013 May 24];44 Suppl 1:S94-102. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/21888989>
6. De Moor G, Sundgren M, Kalra D, Schmidt A, Dugas M, Claerhout B, et al. Using electronic health records for clinical research: The case of the EHR4CR project. *J Biomed Inform* [Internet]. 2014 Oct [cited 2014 Oct 29]; Available from: <http://www.sciencedirect.com/science/article/pii/S1532046414002263>
7. Turisco F, Keogh D, Stubbs C, Glaser J, Crowley Jr WF. Current status of integrating information technologies into the clinical research enterprise within US academic health centers: strategic value and opportunities for investment. *J Investig Med* [Internet]. 2005 Dec [cited 2016 Sep 22];53(8):425–33. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/16354581>
8. DiLaura R, Turisco F, McGrew C, Reel S, Glaser J, Crowley WF. Use of informatics and information technologies in the clinical research enterprise within US academic medical centers: progress and challenges from 2005 to 2007. *J Investig Med* [Internet]. 2008 Jun [cited 2016 Sep 22];56(5):770–9. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/18525452>
9. Murphy SN, Dubey A, Embi PJ, Harris PA, Richter BG, Turisco F, et al. Current state of information technologies for the clinical research enterprise across academic medical centers. *Clin Transl Sci* [Internet]. 2012 Jun [cited 2013 Sep 28];5(3):281–4. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/22686207>
10. Harris PA, Taylor R, Thielke R, Payne J, Gonzalez N, Conde JG. Research electronic data capture

(REDCap)--a metadata-driven methodology and workflow process for providing translational research informatics support. *J Biomed Inform* [Internet]. 2009 Apr [cited 2013 Sep 20];42(2):377–81. Available from:

<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2700030&tool=pmcentrez&rendertype=abstract>

11. Effort seeks to integrate clinical, research data systems | Research News @ Vanderbilt | Vanderbilt University [Internet]. Available from: <https://news.vanderbilt.edu/2016/01/27/effort-seeks-to-integrate-clinical-research-data-systems/>
12. REDCap Technical Overview. [Internet]. Available from: <https://projectredcap.org/REDCapTechnicalOverview.pdf>