

Increasing Business Impact of Connected Standards with a Metadata-Driven Platform

Sergio Alegria, TrialTwin

José C. Lacal, TrialTwin

ABSTRACT

Regulatory agencies require sponsors of drugs, devices, and veterinary products to comply with specific standards and terminologies for all submissions. There are multiple Standards Organizations that develop and curate multiple data standards. Standards are available in many sources and formats, creating a complex ecosystem in the communications between Regulatory Agencies, Standards Organization, sponsors and CROs. While standards are a central piece in an organization's operations, the management of said standards is a complex task that impacts the process of bringing a new product to the market. TrialTwin's Data Standards Governor is a system that allows organizations to manage their own standards and terminologies, having a baseline of pre-loaded standards and terminologies from Standards Organizations (CDISC, LOINC...). The system provides organizations with tools to define connectivity between standards and terminologies to enhance metadata usability throughout the organization.

INTRODUCTION

Regulatory submissions are required for all products developed by sponsors of drugs, devices, and veterinary products. Each agency has a set of standards and terminologies that submissions must comply with.

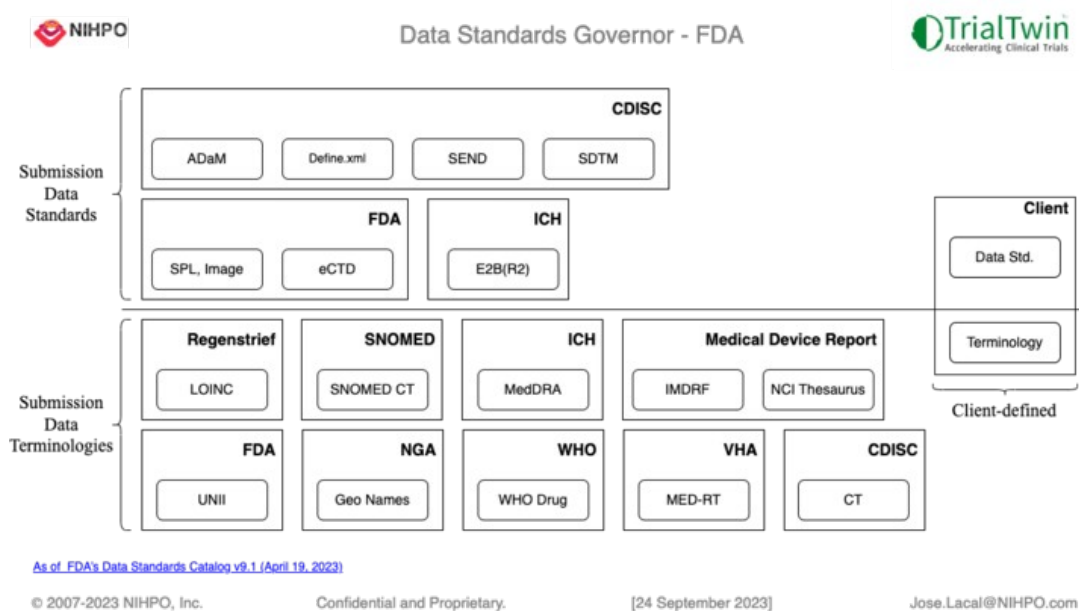
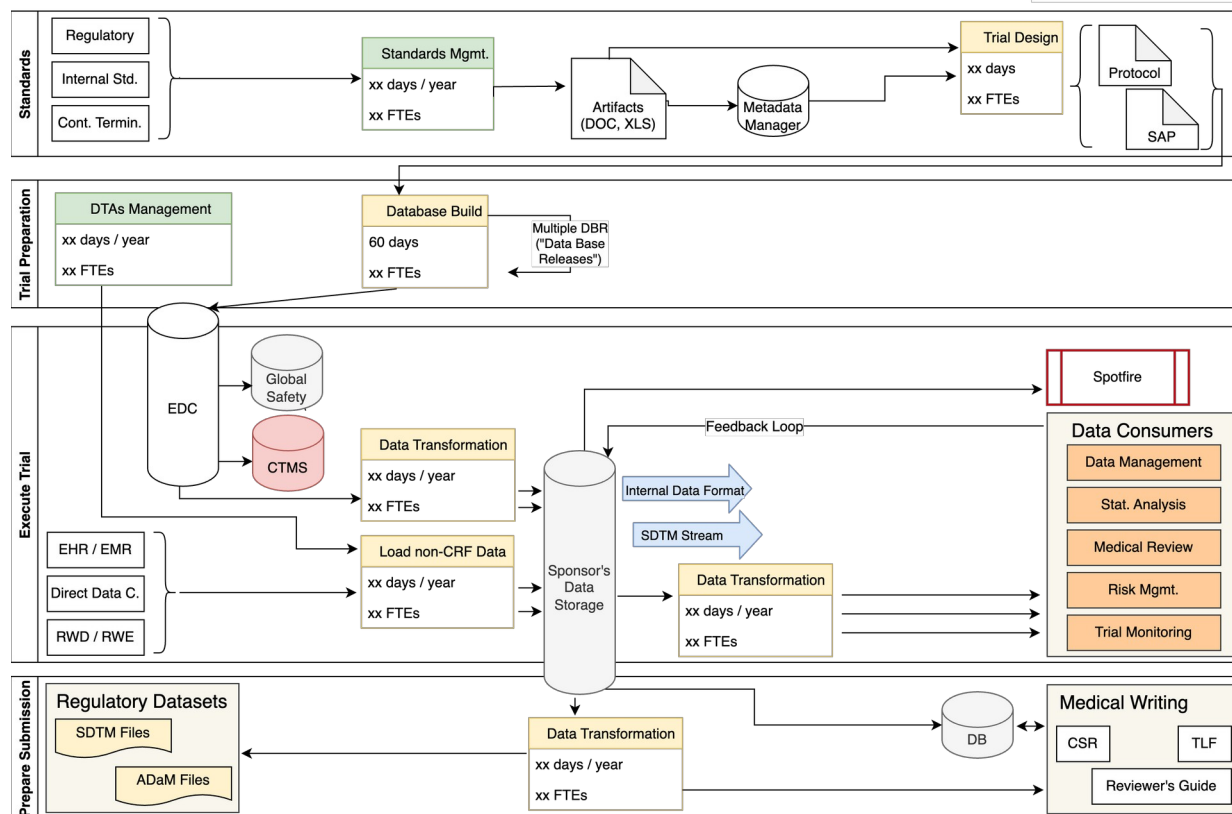


Figure 1: Diagram 1. FDA Data Standards and Data Terminologies Catalog

Figure 1 shows all data standards and terminologies defined in the FDA's Data Standards Catalog, this document contains the supported data standards and terminologies for any regulatory submission to the agency. While many of the standards and terminologies in the catalog are maintained by Standard Organizations, sponsors can include client-defined Data Standards and Terminologies in their submissions.

It is common for Sponsors to define their own data standards that better align with the workflows within the organization. These sponsor defined data standards are usually supersets or variations of standards defined by a Standard Organization, such as CDISC or LOINC among others. These standards are generally used in the processes involved in the development of a new product, but submissions to regulatory agencies are performed using those standards the regulatory supports. Therefore creating the necessity for data transformation processes that differentiate the sponsors operations to the data used in submissions.



© 2007-2023 NIHPO, Inc.

Confidential and Proprietary.

[09 June 2023]

Jose.Lacal@NIHPO.com

Figure 2: Clinical Trial Data Handling

Figure 2 describes the data handling processes involved in a clinical trial from the beginning to submission. The first step when developing a clinical trial is defining the protocol, and to do so organizations usually leverage metadata from previous clinical trials. After the protocol is specified, the data collection processes are defined. The structure of the data that is going to be collected for analysis, submission, medical review, and other data consumers, is of vital importance. If the metadata were to be modified after the data collection has started may have a great impact in the clinical trial timelines. The data collection involves Clinical Research Forms (CRFs), vendor's data, real world data, real world evidence, and other data sources that may be added. Having multiple data sources implies handling different structures of data that need to be transformed and aggregated. These transformations are mappings between different metadata, or data structures. If standards connectivity were to be defined in the Metadata Manager, the development of these mapping processes could achieve a high level of automation.

METADATA MANAGER

Microsoft Excel is a tool used by organization for standards management, creating multiple documents for version control, that are usually shared in a document repository, such as Sharepoint or similar tools. The fact that Excel is the de facto solution to manage standards carries several drawbacks:

- Data standards silos can appear easily as Excel doesn't have a simple way of defining connectivity between different standards.
- The creation of silos is likely paired with different teams dedicated to maintain each silo, and it may create friction points between the steps of the processes.
- When a user needs to search for specific item within a standard, the user will need to identify the files for the version of the standard, and the file containing the valid state at the date where the search must be applicable.
- Maintaining standards up to date can become a time consuming task that requires a large amount of manual effort.

DATA GOVERNANCE

Google defines data governance as: "Data governance is everything you do to ensure data is secure, private, accurate, available, and usable. It includes the actions people must take, the processes they must follow, and the technology that supports them throughout the data life cycle." (What is Data Governance?, n.d.)

DATA STANDARDS GOVERNOR

TrialTwin's Data Standards Governor (DSG) is a system to store and manage data standards and terminologies with built-in governance capabilities. The system has been designed to solved the drawbacks mentioned in the Metadata Manager section in the introduction. The DSG platform is designed with an API first approach, allowing for integrations with other

systems, and for the development of automation workflows. Allowing organizations to have a single source of truth for data standards and terminologies would increase the usability of standards, and would largely reduce the time spent by users searching content, therefore accelerating all processes that may be blocked by a metadata related issue.

FLEXIBLE STORAGE

The DSG has multiple level of storage that can be created:

- **Namespaces** are groups of collections that must have a unique name within them.
- **Collections** are a versioned group of items. Collections define a hierarchy within a namespace, thus allowing to model dependencies within the data managed in the system.
- **Items** are the smallest unit of data that can be managed in the system. Every item is defined in a collection's version.

The system allows all data units defined above to be created by users with enough permissions, thus providing the flexibility to accommodate different data models. This level of flexibility when managing content within the system allows organizations to define the schema that best fits their existing processes.

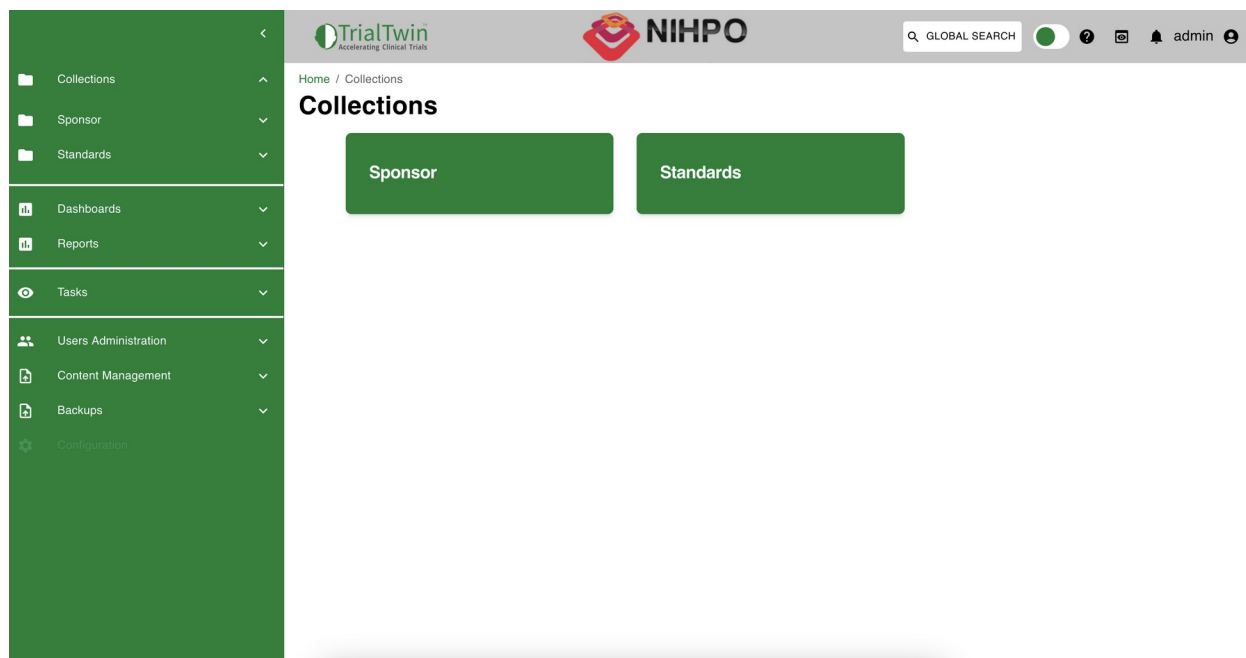


Figure 3: DSG Namespaces

The Figure 3. shows the namespaces page in the TrialTwin Data Standards Governor. There are two namespaces loaded for the examples that will be used throughout this paper:

- Sponsor namespace contains sponsor defined standards modeling CRFs library, and Studies being a variation of the CRFs defined in the library.
- Standards namespace contains CDISC defined standards and terminologies.

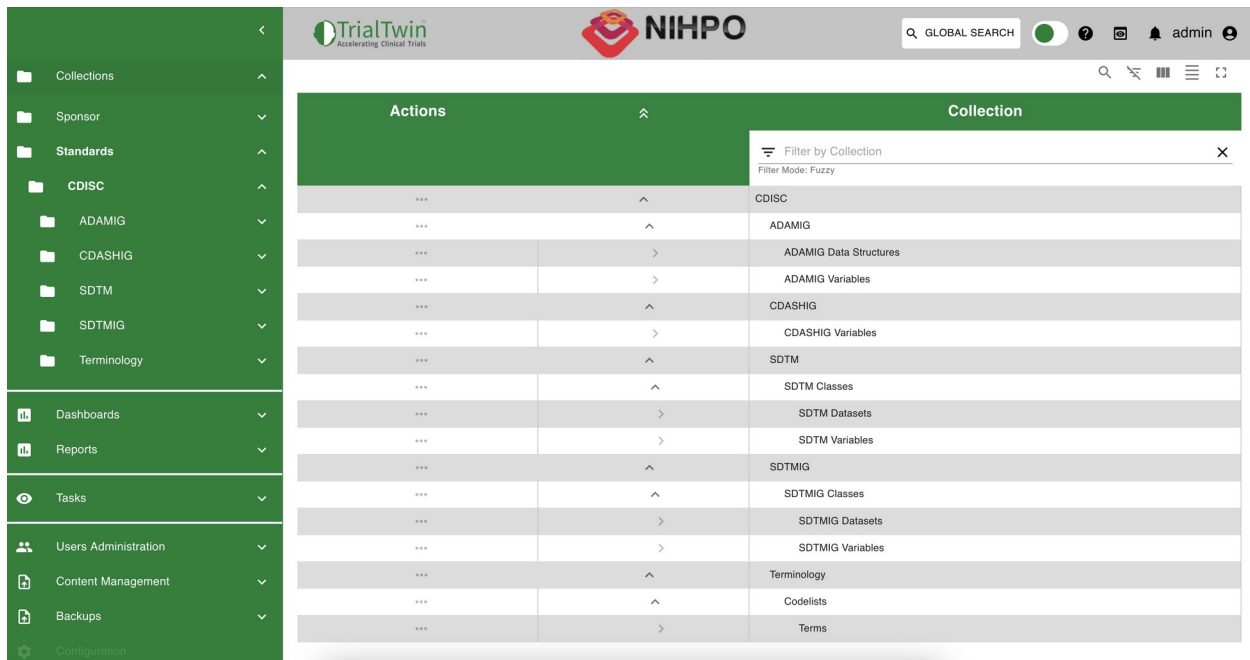


Figure 4: DSG Standards Namespace Collections

Figure 4 shows a listing of the collections loaded in the Standards organization. The table present a hierarchy with the pre-loaded CDISC standards provided by default in the system.

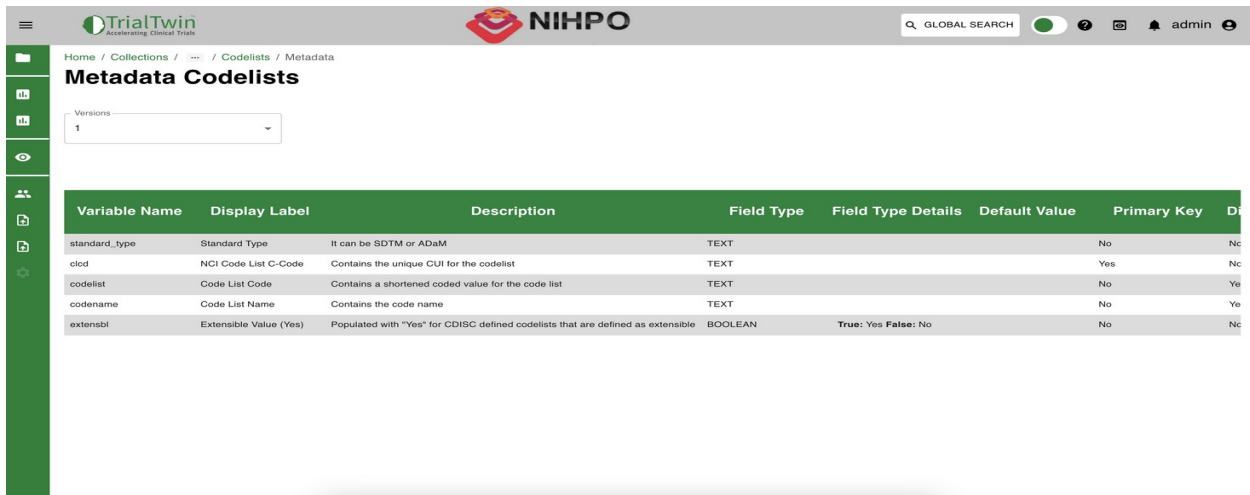



Figure 5: Codelists Metadata


Figure 5 shows a the list of attributes defined to model the Codelists. There are four columns created based on the CDISC Terminology, and a Standard Type collection that stores either SDTM or ADaM, as those are the SDTM Terminologies loaded for this examples. This metadata defines the variables that will be stored for Items within this Codelists version 1, this collection may not have different version, as CDISC Terminologies are not versioned as other standards defined by this organization, such as SWhen defining the metadata for a collection version, the system creates an SQL table based on the attributes definition. Those attributes are defined by the following fields:

- Variable Name: defines a unique name in a Collection that identifies the attribute.
- Display Label: text used in the User Interface when referring to this attribute.
- Description: text describing the attribute purpose.
- FieldType defines the type of attribute. The supported itemTypes are divided in two groups:
 - Natively supported in PostgreSQL: text, integer, numeric, boolean, datetime, list, map, json.
 - System defined types:
 - Derivations define rules that compute an attribute at the item level based on other attributes of the item.
 - Collection store a reference to an item in a different collection. This fieldType is explained in more detail in the Connectivity section of this paper.
 - Inherited attributes are values that are imported from the parent item. This fieldType is only available in collections that are children collections.
 - Sequences are managed at the namespace level, and generate a unique value. The behavior of the sequences can be defined to create a numeric value, or to append a text prefix and/or suffix to the generated value.





- **CachedSequences** are sequences that require a key value to be generated, if multiple requests to generate a value for the same key the response of the sequence will always have the same value.
- **FieldTypeDetails** stores the configuration for the fieldTypes that require it, for instance sequences.
- **DefaultValue** stores a value that will be populated by the system when no value is provided.
- **PrimaryKey** defines whether this attribute participates in the set of attributes that identify the unicity of an item in the collection.
- **DisplayAttribute** defines whether the attribute is used in the User Interface instead of showing the system generated ids.
- **GlobalSearch** defines whether the attribute is included in the GlobalSearch lookup.
- **NotNull** defines whether the attribute is required.
- **HideFromUser** defines whether the column visibility is set up by default in the Search interface in the User Interface.



TrialTwin
Accelerating Clinical Trials



NIHPO

[Home](#) / [Collections](#) / [Terms](#) / [Metadata](#)

Metadata Terms

Versions

1

Variable Name	Display Label	Description	Field Type	Field Type Details	Default
codelist	Code List Code	Contains the code name	INHERIT	Parent Name: codelists Parent Version: 1 Variable: codelist	
codename	Code List Name	Contains the code name	INHERIT	Parent Name: codelists Parent Version: 1 Variable: codename	
ncitrmcd	NCI Code C-Code	Contains the unique CUI for the codelist	TEXT		
subm_val	Submission Value	Contains the submission value which needs to be included in the submission data sets	TEXT		
synonym	Submission Value Synonyms	Contains synonyms	LIST	List Type: TEXT	
definition	Submission Value Definition	Contains the definition of the submission value	TEXT		
nciptrm	NCI Preferred Term	Contain the NCI preferred term for the submission value when available	TEXT		
cldc	NCI Code List C-Code		INHERIT	Parent Name: codelists Parent Version: 1 Variable: cldc	

Figure 6: Terms Metadata

Figure 6 shows the attributes defined to store the Terms from the SDTM Terminology. There are multiple attributes which are inherited from the parent, in this case the Codelists, if there is a change the parent item the system will add an entry to the history of the Term item applying the change in the values from that attribute, therefore ensuring data consistency while maintaining the history of every item.

VERSION CONTROL

Every item stored in the system is assigned a unique id that is used to keep track of the item's history. Two attributes `val_from` and `val_to` are added to every record stored in the database, containing the timestamps in which the record was made valid and when the record was retired.

Actions	#	NCI Code C-Code	NCI Code List C-Code	Valid From	Valid To
Filter by NCI Code C-Code Filter Mode: undefined	1	C102111	C118971	11 October 2023	Current
Filter by NCI Code List C-Code Filter Mode: undefined	2	C102111	C118971	1 January 2023	11 October 2023

Figure 7: Item History

Figure 7 shows the history of an item, a term, the first valid state of the item was defined the first of January 2023, there was an update to this item that created a new entry populating the valid_from timestamp, and filling the valid_to timestamp in the previous valid record.

LINKAGES ACROSS STANDARDS AND TERMINOLOGIES

Defining connectivity between the data standards and terminologies allows organizations to enhance standards utilization. The system allows to define attributes that store a reference to items in other Collections, standards or terminologies.

Figure 8 shows the metadata used to model fields in a CRF. There is a variable, `cdisc_fields_target`, that stores a reference to the SDTMIG variable defined by CDISC in the SDTMIG

Actions	Variable Name	Display Label	Description	Field Type	Field Type Details
	<code>cdisc_fields_target</code>	CDISC FIELDS Target	Target from the standards in CDISC CT	COLLECTION	Collection: sdtmig_variables Version: 3.4 Display Variables: variable_name
	<code>field_oid</code>	Field OID	Field Name	TEXT	
	<code>field_name</code>	Field Name	Name of the field	TEXT	
	<code>ordinal</code>	Ordinal	Ordinal of the field	INTEGER	
	<code>data_format</code>	Data Format	Data format of the field	TEXT	
	<code>dictionary_name</code>	Dictionary Name		COLLECTION	Collection: onco_dictionaries Version: 1 Display Variables: dictionary_name
	<code>unit_dictionary_name</code>	Unit Dictionary Name	Name of the unit dictionary which controls the units of the field	TEXT	

Figure 8: CRF Field Metadata

3.4. Defining this attribute enables sponsors use the SDTMIG variable as baseline for the variable being modeled in the CRF field. The collection fieldType can be customize to store a reference to items that are children of other item. If an attribute can only be populated with values that are children of a Codelist, the system allows to store a Collection fieldType that references to Terms, selecting a parent Codelist, creating a constraint that validates that the values populated in that variable, are valid Terms whose parent is the defined codelist.

This references allow organizations to build their standards based on content that is managed within the system. When any item is modified, all item that are connected to it will be impacted.

Collection	Related Entry	Relationship
CodeLists	CCCATICategory of Clinical Classification	Parent
Onco Dictionary Entries	Abnormal Involuntary Movement Scale Clinical ClassificationC118971	Pointed By
Cardio Dictionary Entries	Abnormal Involuntary Movement Scale Clinical ClassificationC118971	Pointed By

Figure 9: Item Connectivity

Figure 9 shows a table containing all relationships of an item. The system manages two type of relationships, four taking the direction into account:

- Parent-Children is the relationship created between items when the Collections they are created in have a parent-children relationship defined in the hierarchy.
- Collection fieldType define are dependencies of Items that store references to other items. If item X stores a reference to Item Y. In the relationships table of Item X the relationship will be Pointing to; while in the same table for Item Y it will be Pointed By. This allows to analyze the impact of changes in the system before modifying any data.

Being able to define connectivity between standards, it would possible to implement a standards-driven trial data management model, in which the identification of which terminologies are used by the CRFs will be stored in a connected single source of truth, that would allow users to identify dependencies within a study.

CONTENT IMPORTER

The system has built in content importer capabilities, that allow to load content into the system from Excel files. Being able to load data from Excel, allows users to quickly load any data that is already available within the organization, thus allowing a fast and simple set-up and migration to use the system.

The system provides with a Staging area in which the files are loaded, both the structure and the data are validated. There are some validation steps that run for all collections and versions: variable_name matching and no duplicate records based on primary keys, and no duplicate rows in the system, to avoid duplicates when no primary key is defined. Other validation steps run based on the definition of the metadata: when a Collection fieldType is defined the system validates that the record exists in the system; non nullable attributes validate that the variable is populated.

Actions	Collection	Version	Ran Validation	Is Valid	Updated On	Locked
...	Onco Forms	2	Yes	No	13 October 2023 at 06:23 (UTC)	No

Figure 10: Content Importer - Staging Area

Figure 10 shows a list of all files loaded in the Staging Area, the validation run details, and an actions column. The possible actions are: review validation in the app, download a validation report as an Excel file, and to commit the changes, that is only available if the content is valid.

Actions	Row	formoid	Name	question	Validation Duplicate Row
	1	SDF01	Form01	Form01 question	Yes
	2	SDF20	Form20	Form20 question	No
	3	SDF21	Form21	Form21 question	No

Figure 11: Validation Report

Figure 11 shows a validation report that attempts to load three records, one of which has a repeated value in the primary key field. The system runs the validation in the staging area, and does not allow to commit the file until the validation run is successful. The actions column allows to modify the content of each row, the user can save the changes, and the validation will run automatically. These process must be repeated until all the content is validated, and the content can be committed. Committing the changes will not alter the content within the system directly, it will kick off the governance process to alter the system state.

GOVERNANCE

The system has governance capabilities that are automatically created for all actions that have impact in the data. When a user performs an action, a Task is created, and appended to the TaskPool that contains all tasks that have not been resolved. All actions in the system have assigned a scope, users have access to actions based on the scopes they are granted by the different roles that can be set up by the system administrators.

Task resolution is achieved after going through a governance workflow involving multiple users. By enforcing at least two different users to actively participate in a Task resolution, the system is restricting the ability of a single user to modify the state of the system.

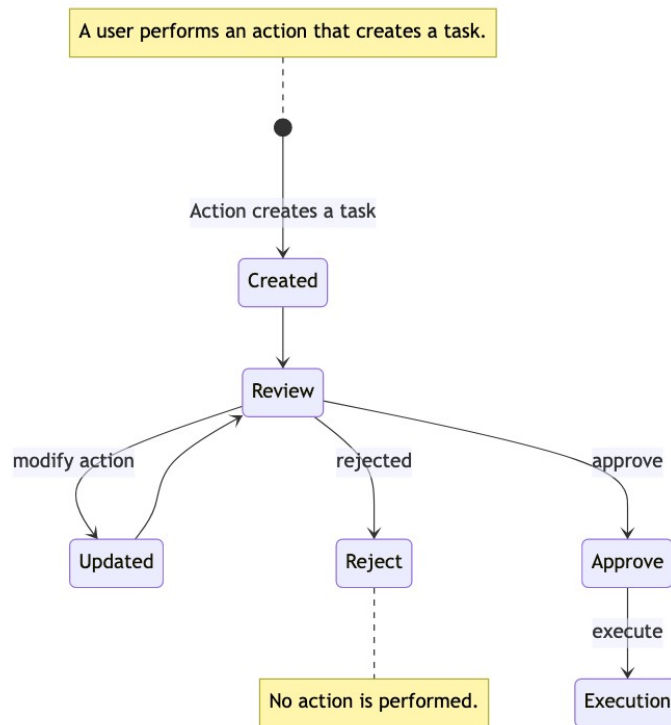


Figure 12: Task's Status Diagram

When a user, User A, with permissions to perform an action in the system a task will be added to the TaskPool, thus kicking off the governance workflow for that task. When a second user, User B with enough permissions to access the tasks, selects the task from the TaskPool, there are three possible operations:

- Updating the action that was originally proposed. For instance if the task contains an action to create an item, User B could modify the attributes that were first proposed by User A. After the task is updated, User B will no longer have access to the task in the TaskPool, as they are the last user to modify the task.
- Rejecting a task closes the task without performing any action.
- Approving a task will execute the action as defined in the Task. The system will show an error to the user in case the action could not be executed properly, if the task executes with no errors the system will show a message to the approver user.

Tasks can define one or more actions, in that case when a task is approved all changes must execute successfully. If an error arises in any action, no action from the task will be committed, and the system will remain unaltered.

Whenever a user performs one of the above mentioned operations in a task, a message is required. By doing so the system stores the history of every task. These entries contain which user performed the action, a timestamp, and a message.

AUDITING

Auditing provides organization with a detailed audit trail for tracking and validation purposes. Being able to track all changes that are performed to the content managed in the system. Tasks contain all history of actions, and changes that have occurred in the system. It is possible to know when an item was first created along with its history. Which users were involved in any of the modifications the item went through, and timestamps are stored to provide the ability to identify when the action was attempted.


```

"history": [
  {
    "message": "Remove source from help_text",
    "user": {
      "user_uid": "5614bab4-5968-4343-bb38-c5cccf8686a7",
      "email": "user-admin@example.com"
    },
    "status": "created",
    "updated_on": "2023-10-13T06:49:38.060972"
  },
  {
    "message": "Modify comment field value.",
    "user": {
      "user_uid": "162f5e3b-f675-48e9-bf87-7cf011973a56",
      "email": "admin@example.com"
    },
    "status": "updated",
    "updated_on": "2023-10-13T06:51:45.658950"
  },
  {
    "message": "Comment should include source reference.",
    "user": {
      "user_uid": "8d0a0c9b-1ae5-4aba-9ac7-18abe7aba6ff",
      "email": "approver@example.com"
    },
    "status": "rejected",
    "updated_on": "2023-10-13T06:52:57.884694"
  }
],

```

Figure 13: Task's History

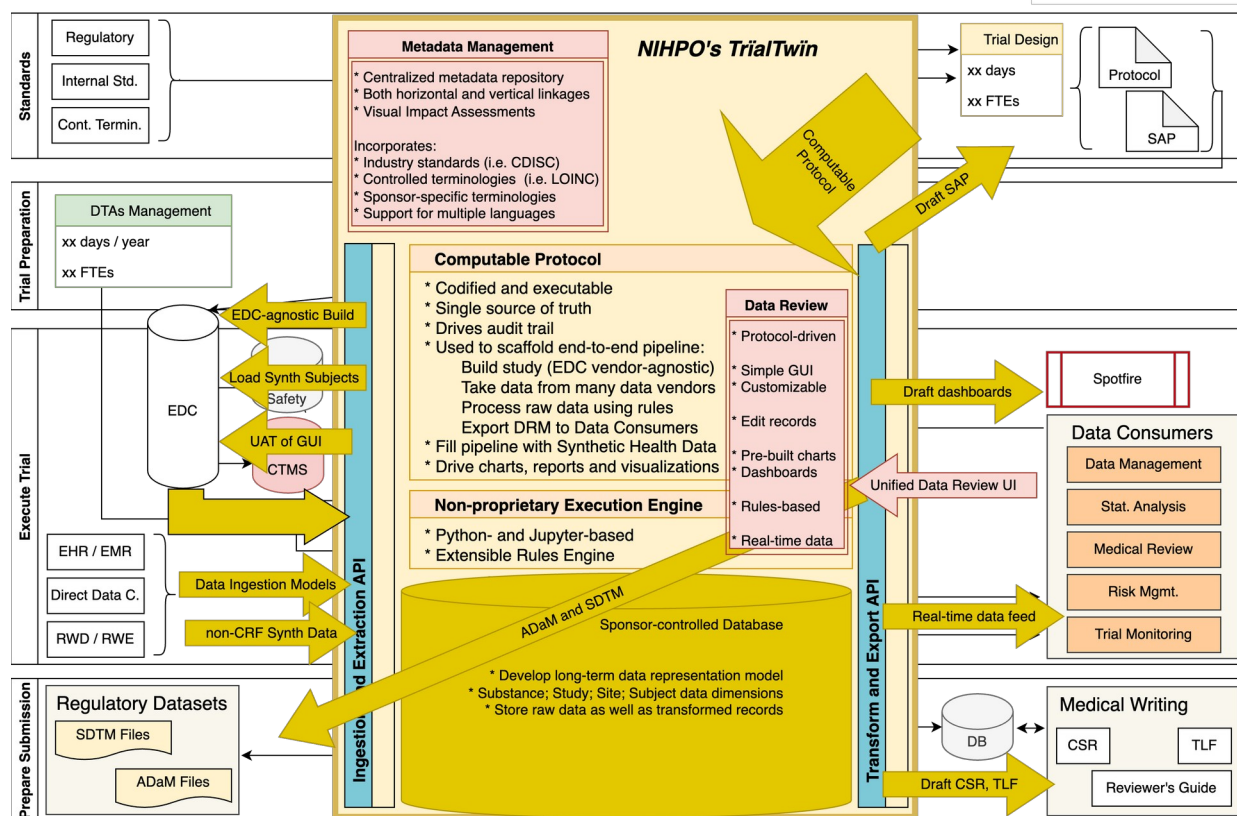
Figure 13 shows the history of one task. This task went through three steps performed by three different users. It was first created with an associated change, not shown in the figure. Then an update to the content was performed. Lastly the task was resolved by a third user with a rejected status.

FUTURE OF THE SYSTEM

This paper covers the current state of the TrialTwin Data Standards Governor, the system is under on-going development, and there are multiple features and enhancements yet to be developed.

Changes that are expected in the near future are:

- Allow customization of governance workflow. In the Governance section this paper has covered the current workflow, and likely it is not aligned with most organizations, we want to provide organizations with flexibility to best fit the system within their current workflows.
- Enhance content importer to support other file formats such as: CSV, JSON, and SAS datasets.
- Enhance the Validation Engine to allow users to define validation rules per collection and namespace. Allowing customization in the Validation processes the system runs before performing any actions.
- Enhance API usability, even further, to simplify integrations with the system.



© 2007-2023 NIHPO, Inc.

Confidential and Proprietary.

[09 June 2023]

Jose.Lacal@NIHPO.com

Figure 14: TrialTwin Data Handling

Figure 14 shows TrialTwin platform, while the Data Standards Governor is a system on its own, it can be integrated with other systems and services from the TrailTwin platform.

TrialTwin also offers the ability to generate synthetic data. The SynthData packages can be used as test data. Having access to the standards and terminologies used for a study, the generation of the SynthData will have the same structure as the data that will be collected once the clinical trial starts. Having access to large amounts of data before the start of the trial, can accelerate the development and QA of processes that depend on data collection. For instance the development of analysis can use a SynthData package that is structurally equal than the data that will be used in the actual analysis to start developing TLFs earlier.

REFERENCES

1. (What is Data Governance?. n.d.) <https://cloud.google.com/learn/what-is-data-governance>

ACKNOWLEDGMENTS

We are grateful to all of those with whom we have had the pleasure to work during the development of this system.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sergio Alegria

TrialTwin

Work Phone: +34 622034131

Email: sergio.alegria@datasdr.com

Web: <https://www.linkedin.com/in/sergio-alegria-548245236/>

Brand and product names are trademarks of their respective companies.