## **Supplemental Figures**

Supplemental Figure 1: Searching and viewing schemas from schema.org and elsewhere A. Visualizing a json schema file - The schema viewer allows users to visualize json schemas when provided a link to the json schema file

	Wha Register Schema	t would you like to do? P Create Schema
WHAT CAN I DO HERE?		Visualize → Register Your Schema
		https://raw.githubusercontent.com/outbreak-info/outbreak.info-resources/master/ Let's Go!
		Examples: BioLink Schema CTSA Dataset Schema

B. Searching/Viewing for a schema in the registry - The Schema registry allows users to browse and view available schema classes so users can find and reuse relevant schemas with ease

	Schema Registry	
SHORTCUTS	Visualize 👁 Schema.org) Visualize 👁 BioLink) Extend 🗜 Dataset	
38 Results	Dataset	<b>b</b>
Compare (Ma	x 4) Details   V validation available	View/Extend
□ ()	Dataset () 25 Properties []   A: cvisb-dataset   Subclass of: schema:Dataset	۲ ۲
☑ ‡	Dataset () 118 Properties E   A: n3c   Subclass of: schema:Dataset V	•
	Dataset 123 Properties 1   A: outbreak   Subclass of: schema: Dataset	4 0

C. Comparing schemas from different sources - If there a similar class exists across multiple schemas, the user can compare the properties that are available (Compare All) or have validation (Compare Used) across the the different schemas

Compare Schemas					
Compare All Properties (Extended and Inherited)		ed and Compare U	Compare Used Properties By Schema (If Specified)		
	ited from Schema.org xtended Definition Compare All	31 1	Only properties used as specified in validation (embedded JSON-schema validation) E Compare Used		
35 properties were compared					
Property	▲ n3c:Da	taset 🔺	outbreak:Dataset		
keywords		~	~	^	

Compare Schemas

X

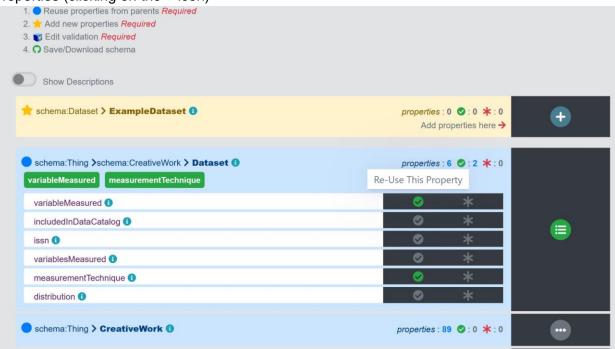
Reywords		
license	~	~
measurementTechnique	~	✓
name	~	~
release_frequency	~	×
standards_used	~	×
url	~	×
citedBy	×	~

Supplemental Figure 2 - Extending and customizing a pre-existing schema for a particular use A. Extending/Tailoring a schema from schema.org - If a suitable class for reuse is

identified, it can easily be customized by clicking on the 'extend class' icon



B. Selecting existing properties to inherit/include for validation - To customize a class, the user simply selects properties to be inherited (by clicking on the check icon) or creating new properties (clicking on the + icon)



C. Creating new properties - New properties are directed towards conforming to basic schema.org conventions via the property creation guide

a:Dataset > ExampleDataset ()	properties : 0
New Property	
Name ≭	
curatedBy	E3
Learn about naming conventions here	
Description ≭	
The source of the Dataset. Provides provenance for data	asets curated by others.
Domain <del>×</del>	
ExampleDataset	
Input Type(s) ★	
schema:Organization ×	
Dataca	ADD
schema:DataCatalog	

Supplemental Figure 3 - Creating validation for the schema for data quality enforcement

A. Including Marginality - Marginality is enforced in the validation by toggling whether or not a property is required via the \* icon

schema:Thing >schema:CreativeWork > Dataset  variableMeasured measurementTechnique	properties : 6 오 : 2 ★ : 1 Mark As Required
variableMeasured ()	⊘ *
includedInDataCatalog 0	<u> *</u>
issn 🚺	
variablesMeasured 1	⊘ *
measurementTechnique 🕕	⊘ *

B. Toggling the Validation Editor - Validation can be further customized via the Validation Editor



C. Accounting for Cardinality through the validation editor - Cardinality (one/many) is handled via the validation by selecting and editing the validation option. Single options like string or url would be equivalent to a cardinality of one, while an option like string(s) would be equivalent to a cardinality of many (as seen in figure). A cardinality of many for a type other than string can be attained by creating a new validation option (using oneOf or anyOf) or by editing the string(s) option (which uses oneOf). Once the custom validation is saved by clicking on the green 'save' button, it will be an option which can be drag/dropped to the appropriate property



D. Customizing a common validation option - Selecting the (+) option will open a blank editor which will enable the creation of new validation options. To create an organization validation option for the curatedBy property, the content of the object|Person option was copied into a new validation option and edited to replace the "@type" : "Person" with "@type" : "Organization". Once saved, the option becomes available for drag/drop

Your Validation	Common Validation Options
<pre>curatedBy ③ {     "description": "The source of the } variableMeasured ⑧     "description": "The variableMeasure </pre>	Trag & drop to merge validation         string (2)       tring(s) (2)         integer (2)       ull (2)         ull (2)       teges (2)         date (2)       (2)         object   Person (2)       (2)
measurementTechnique	<pre>1 { 2 "@type": "Organization", 3 "type": "object", 4 "properties": { 5 "name": { 6 "type": "string" 7 }, 8 "url": { 9 "type": "string", </pre>
	<pre>10 "format": "uri" 11 } 12 }, Save</pre>

Supplemental Figure 4 - Exporting and saving a schema generated by the Schema Playground editor

A. Exporting/Downloading your schema locally - The schema can be downloaded locally using the download icon

Validation Editor			
Download your schema Define how the input for each property should be validated, drag and drop common rules or create your own.			
Your Validation	Common Validation Options     Drag & drop to merge validation		
<pre>curatedBy ③     {         "description": "The source of the         "@type": "Organization",         "type": "object",         "properties": {              "name": {                  "type": "string"               },              "url": {                 "type": "string",</pre>			

B. Saving your schema to Github - If the user would like to register and share the schema, then it can be saved to a Github repository

😂 🚸 🚺 🖓 🕯	Validation Editor	
Save (Requires GitHub Logi	<mark>n)</mark> the input for each property should be	be validated, drag and drop common rules or create your own.
	alidation	Common Validation Options Drag & drop to merge validation
"@ "t	escription": "The source of the type": "Organization", ype": "object", roperties": { "name": { "type": "string" }, "url": { "type": "string",	string     string
	escription": "The variableMeasu neOf": [ { "type": "string"	

C. Customizing your commit of the schema to Github - When saving to Github, the user can select from their pre-existing repos by clicking on the 'Get Repos' button which will populate the available repos dropdown selector. Once a repo is selected, the user can either override an existing file or create a new file. The user can also customize the git commit message/comment

		2		
🐱 Save your work to GitHub				
<ul> <li>By continuir</li> </ul>	g with this process you are giving us permission to access your GitHe	ub account to		
create/edit pub	lic repos/files. If you DO NOT accept this please close this window a	nd choose the		
Download opti	on to save manually.			
1. Choose a repository	2. Choose file	3. Save to GitHub		
Choose an existing repository to save your work to.	Create or update an existing file. Do not include the file extension. It will be <code>.jsonld</code> .	A new file will be saved to this repository.		
1. Choose one (All available):	Create File	Create:		
		DDE_schema_tests/ exampleDataset ?		
DDE_schema_tests \$	· · · · ·			
Search by name:	A. New File: Name your file (without extension)	C Save		
· · · · · · · · · · · · · · · · · · ·	exampleDataset			
	B. (Optional) Write a comment about this addition.			
	github commit comment goes here			
	b	1		