SchemaBlocks {S}[B] | GA4GH Connect | Michael Baudis | 2021-03-04

General (Re-)Introduction
“cross-workstreams, cross-drivers” initiative to document GA4GH object standards and prototypes, data formats and semantics

- launched in December 2018
- documentation and implementation examples provided by GA4GH members
- no attempt to develop a rigid, complete data schema
- object vocabulary and semantics for a large range of developments
- currently not “authoritative GA4GH recommendations”
- recognized in GA4GH roadmap as element in "TASC" effort

schemablocks.org
**SchemaBlocks Home**

SchemaBlocks is a "cross-workstreams, cross-drivers" initiative to document GA4GH object standards and prototypes, as well as common data formats and semantics.

Launched in December 2018, this project is still to be considered a “community initiative”, with developing participation, leadership and governance structures. At its current stage, the documents can not be considered “authoritative GA4GH recommendations” but rather represent documentation and implementation examples provided by GA4GH members.

While future products and implementations may be completely based on SchemaBlocks components, this project does not attempt to develop a rigid, complete data schema but rather to provide the object vocabulary and semantics for a large range of developments.

The SchemaBlocks site can be accessed though the permanent link schemablocks.org. More information about the different products & formats can be found on the workstream sites. For reference, some of the original information about recommended formats and object hierarchies is kept in the GA4GH Metadata repositories.

For more information on GA4GH, please visit the [GA4GH Website](https://www.ga4gh.org).

**SchemaBlocks “Status” Levels**

SchemaBlocks schemas ("blocks") provide recommended blueprints for schema parts to be re-used for the development of code based "products" throughout the GA4GH ecosystem. We propose a labelling system for those schemas, to provide transparency about the level of support those schemas have from [5][8] participants and observers.

- **Proposed (5) [8]** Status Levels
  - The current status level of those recommendations is "proposed".
  - early development or import stage, of any quality
  - no recommendation; existence does not mean any current or future (5) [8] support
  - **proposed**
    - at least some (5) [8] contributors are in favour of such a block
    - the code may undergo considerable maturation
    - not recommended for integration into products w/o close tracking
    - contributions and discussions are encouraged

- **Implemented**
  - mature block which is implemented in one or more (5) [8] aligned schemas
  - may be extended from a core block or be too specific for general ("core") usability

- **Core**
  - a schema block with recommended use
  - stable through minor version changes
  - has to be used in at least 2 standards/products approved by the GA4GH Steering Committee

**SchemaBlocks (5) [8] Mission Statement**

SchemaBlocks aims to translate the work of the workstreams into data models that:

- Are usable by other internal GA4GH deliverables, such as the Search API.
- Are usable by Driver Projects as an exchange format.
- Aid in aligning the work streams across GA4GH.
- Do not create a hindrance in development work by other work streams.

After discussions with stakeholders from GA4GH work streams and driver projects who create data models (such as Phenopackets, Search API) or who would use SchemaBlocks for the development of their APIs and data exchange formats (Beacon, EGA, Gel), the SchemaBlocks team has come up with the following principles for this initiative:

**Work Stream Interactions**

Work streams will continue to create standards proposals and their own coherent project implementations, but will work with the SchemaBlocks group to write the Blocks that will come from their own work and are considered of overarching use. Generally, primary work stream and driver project outputs will live in their own spaces outside of SchemaBlocks, with shareable, mature elements - code, documentation, implementation snapshots - being represented in (5) [8].
{S}[B] As a Component Recycler

"Donor" Schema

Root:
{
  Components: {
    this: {...},
    that: {
      $ref: './Other'
    }
  }
}

Other:
{
  Components: {
    well: {...},
    done: {...}
  }
}

{S}[B] Representation

Other:
{
  Meta: {
    donorSchema: "DonorProject",
    donorLink: "donorschema.org",
    status: "core"
  },
  Components: {
    well: {...},
    done: {...}
  }
}

"Acceptor" Schema

Root:
{
  Components: {
    my: {...},
    theirs: {
      $ref: "schemablocks.org/other.json"
    },
    more: "..."
  }
}
Dissection & Transformation

Use Case Transforming Phenopackets objects (here "Age") into JSON Schema documents with (proposed) stable id and address as well as "human readable" documentation & examples.

- Excerpt from Phenopackets v1.0 Schema
- written in Protocol Buffers (Google's data serializing format)
- separate documentation rendered in "ReadTheDocs"
• Separate (S)[B] repository for parental project
• here "sb-phenopackets"
• individual schema documents for each original object
• (currently) manual re-write into JSON Schema documents (YAML version), including metadata header (id, provenance ...)
• versioned
Dissection & Transformation

• schema documents are programmatically converted into different outputs
• a Markdown document with "Jekyll" header is auto-converted by Github into a complete website document, including inline code examples
Dissection & Transformation

• schema documents are programmatically converted into different outputs
• a versioned JSON document serves as canonical reference for integration into other products/schemas
### {S}[B] JSON Schemas

(versioned & current)

- Schema definition documents (YAML) are rendered into JSON Schema `.json` documents.
- The latest version is kept under "current", with parallel versioned & archived copy & original YAML, JSON in source repository.

https://schemablocks.org/schemas/ga4gh/current/DataUseLimitation.json

https://schemablocks.org/schemas/ga4gh/v0.0.1/DataUseLimitation.json


---

### DataUseLimitation

**Class**: `DataUseLimitation`

**Type**: object

**Description**: The [GA4GH] Data Use Ontology (DUO) includes terms describing data use conditions, in particular for research data in the health/clinical/clinical domain.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>string</td>
<td>Free text description mainly to encapsulate those conditions that wouldn't be described by DUO terms or modifiers.</td>
</tr>
<tr>
<td>modifier</td>
<td>boolean</td>
<td>Modifier/restriction applicable for the DUO.</td>
</tr>
</tbody>
</table>

#### Examples

```json
{
  "dataUseLimitation": {
    "modifier": true,
    "description": "Free text description mainly to encapsulate those conditions that wouldn't be described by DUO terms or modifiers."
  }
}
```
**BeaconAlleleRequest beacon**

<table>
<thead>
<tr>
<th>[URL] Status</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provenance</td>
<td>Beacon-API</td>
</tr>
</tbody>
</table>

**Contributors**
- Marc Tune
- Michael Board
- Tabea de la Torre Perias
- Joel Faramo
- Beacon-developers...

**Source**
- raw source [SSON] | Github

**Attributes**
- Type: object
- Description: A beacon request as interpreted by the beacon.

**Properties**
- Property: alternateNames
  - Type: string

- Property: assemblyId
  - Type: string

- Property: assemblyName
  - Type: string

- Property: datasetIds
  - Type: array of string

- Property: id
  - Type: integer

- Property: manifest
  - Type: string

- Property: referenceAssembly
  - Type: string

- Property: referenceClade
  - Type: string

- Property: start
  - Type: integer

- Property: summary
  - Type: string

**VariantType**

- Property: alternateNames
  - Type: string

- Property: assemblyId
  - Type: string

**VariantValueExample**

- Property: alternateNames
  - Type: string

- Property: assemblyId
  - Type: string

**Cheksum sb-checksum**

<table>
<thead>
<tr>
<th>[URL] Status</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provenance</td>
<td>Beacon-API</td>
</tr>
</tbody>
</table>

**Contributors**
- Marc Tune
- Michael Board
- Tabea de la Torre Perias
- Joel Faramo
- Beacon-developers...

**Source**
- raw source [SSON] | Github

**Attributes**
- Type: object
- Description: A beacon checksum to verify the integrity of the beacon.

**Properties**
- Property: checksum
  - Type: string

- Property: checksumType
  - Type: string

**Biosample sb-phenopackets**

<table>
<thead>
<tr>
<th>[URL] Status</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provenance</td>
<td>Phenopackets</td>
</tr>
</tbody>
</table>

**Contributors**
- GA4GH Data Working Group
- Jules Jacobson
- Peter Robinson
- Michael Baudouin
- Michael Board
- John Luiet
- John Luiet

**Source**
- raw source [SSON] | Github

**Attributes**
- Type: object
- Description: A Biosample refers to a unit of biological material from which the substrate mAbs can be generated.

**Healthy mapping**

- Property: Specimen

**Properties**
- Property: property
  - Type: string

**Checksum**

<table>
<thead>
<tr>
<th>[URL] Status</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provenance</td>
<td>Ga4gh-CRIS (develop branch)</td>
</tr>
</tbody>
</table>

**Contributors**
- Sushil Varma

**Source**
- raw source [SSON] | Github

**Attributes**
- Type: object
- Description: A checksum to verify the integrity of the biosample.

**Properties**
- Property: checksum
  - Type: string

- Property: checksumType
  - Type: string
SchemaBlocks JSON Schema document format

- **{S}[B]** "blocks" are written in the YAML version of a JSON Schema document format
- convenience choice - flexibility, readability, tooling ...
- **not** implying specific semantics beyond some format conventions - extensible for use-case driven requirements

- the **meta** part (itself defined as a schema "block") contains housekeeping information
  - reference address & version
  - provenance & use cases
  - sb_status about "blessing level"

- the properties part defines the attributes including their description and usage examples
  - descriptions & examples provide the core documentation which is deparsed to the website documents

```json
"$schema": "http://json-schema.org/draft-07/schema#"
"$id": "https://schemablocks.org/schemas/ga4gh/AgeRange/v0.0.1"
title: AgeRange
description: Age range
type: object

meta:
  contributors:
  - description: "Jules Jacobsen"
    id: "orcid:0000-0002-3265-15918"
  - description: "Peter Robinson"
    id: "orcid:0000-0002-0736-91998"
  - description: "Michael Baudis"
    id: "orcid:0000-0002-9903-4248"
  - description: "Isuru Liyanage"
    id: "orcid:0000-0002-4839-5158"
  provenance:
  - description: Phenopackets
    id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  used_by:
  - description: Phenopackets
    id: 'https://github.com/phenopackets/phenopacket-schema/blob/master/docs/age.rst'
  sb_status: implemented

properties:
  start:
    allOf:
      - $ref": https://schemablocks.org/schemas/ga4gh/v0.0.1/Age.json
    description: Age as ISO8601 string or OntologyClass
    examples:
      - age: 'P12Y'
  end:
    allOf:
      - $ref": https://schemablocks.org/schemas/ga4gh/v0.0.1/Age.json
    description: Age as ISO8601 string or OntologyClass
    examples:
      - ageClass:
        id: 'HsapDv:0000086'
        label: 'adolescent stage'
        age: 'P16Y6M'
  required:
  - start
  anyOf:
    - start
    - end
  examples:
    - start:
      age: 'P12Y'
      ageClass:
        id: 'HsapDv:0000086'
        label: 'adolescent stage'
    end:
      age: 'P18Y'
```
SchemaBlocks JSON Schema document format

- {S}[B] "blocks" are written in the YAML version of a JSON Schema document format
  - convenience choice - flexibility, readability, tooling ...
  - not implying specific semantics beyond some format conventions - extensible for use-case driven requirements
- the meta part (itself defined as a schema "block") contains housekeeping information
  - reference address & version
  - provenance & use cases
  - sb_status about "blessing level"
- the properties part defines the attributes including their description and usage examples
  - descriptions & examples provide the core documentation which is deparsed t0 the website documents

SchemaBlocks “Status” Levels

SchemaBlocks schemas ("blocks") provide recommended blueprints for schema parts to be re-used for the development of code based "products" throughout the GA4GH ecosystem. We propose a labeling system for those schemas, to provide transparency about the level of support those schemas have from {S}[B] participants and observers.

Proposed {S}[B] Status Levels

The current status level of those recommendations is "proposed".
- **playground**
  - early development or import stage, of any quality
  - no recommendation; existence does not mean any current or future {S}[B] support
- **proposed**
  - at least some {S}[B] contributors are in favour of such a block
  - the code may undergo considerable maturation
  - not recommended for integration into products w/o close tracking
  - contributions and discussions are encouraged
- **implemented**
  - mature block which is implemented in one or more {S}[B] aligned schemas
  - may be extended from a core block or be too specific for general ("core") usability
- **core**
  - a schema block with recommended use
  - stable through minor version changes
  - has to be used in at least 2 standards/products approved by the GA4GH Steering Committee
The document features a diagram illustrating the repository structure of the `S[B] SchemaBlocks Github Repository`. The structure is hierarchical and includes directories for blocks, playground, sb-phenopackets, sb-other-project, tools, source, working, schemas, and generated. The diagram also highlights repository branches named `ga4gh-schemablocks.github.io`, `v0.8.1`, and `current`. The text mentions conversion/validation tools and website repository (Markdown w/ YAML for Github Pages).
SchemaBlocks {S][B] - Directions & Contributions

• Recognized need of having a set of recommended standards for integrating into product development
  ➡ no need to work through complex standards/projects like FHIR, Phenopackets ...
  ➡ simplification of development

• SchemaBlocks {S][B] to assume strategic position in GA4GH *TASC system
  ➡ Inclusion into product approval processes?
  ➡ Management/Support?

• Wish for participation of (GA4GH affiliated) groups & individuals, to *expose* their standards & products

• Most important role is the *community aspect*, the interactive exchange of concepts, ideas, code, knowledge, resources ...

• Technical to-dos:
  ➡ Lifecycle: Versioning and representation of donor schemas?
  ➡ Development of conversion workflows for updated source products?
  ➡ Alternative/conflicting blocks...: Graded recommendations? Name spacing?

*Technical Alignment Sub Committee*
SchemaBlocks {S}{B} in 2021
Establishing as key component of the GA4GH standards ecosystem

• Recognized need of having a set of recommended standards for integrating into product development
  - no need to work through complex standards/projects like FHIR, Phenopackets ...  
  - simplification of development  
  - demonstration of practical usability of referenced {S}{B} schemas in "acceptor" schemas

• establishing stable repository structures, canonical endpoint URLs & formats
• participation of more (GA4GH) groups, to expose their standards & products
• GA4GH procedures - blessed schema status labels?
• support structure - maintenance, workflows, stewardship, versioning / updating ...
• tooling / libraries
SchemaBlocks {S}[B] | GA4GH Connect | Michael Baudis | 2021-03-04

Tooling
{S}[B] SchemaBlocks Github Repository Structure

YAML > Perl > JSON & MD > Jekyll > HTML
{S}[B] Tooling

sbSchemaParser

- single tool which provides schema deparsing and output page generation across a local mirror of schema repositories and {S}[B] website source directories => "Q & D"

- configuration file for selection & labeling

  ➡ Works for "all in one" flexible prototyping but should be replaced by modular & safer tools

tools

Tools for managing the {S}[B] repositories and website

sbSchemaParser

The sbSchemaParser.pl script is used to process schema files written in JSON Schema (YAML version) into human-readable documentation (e.g. Markdown files for Jekyll based HTML generation) and JSON data files from the embedded examples.

Directives for source and target directories can be modified in the .config.yaml file in the script's directory. The general repository structures for the repositories which are being parsed by sbSchemaParser is shown below.

{S}[B] Repositories

{S}[B] code repositories adhere a consistent structure & naming:

```
 sb-code  # each of the code repositories
   |  # original code
   |- working  # for editing, temporary...
   |- schemas  # JSON Schema files as YAML; read to produce the output files
   |
   |- generated  # contains files generated from main schema YAML files
   |
   |- examples  # JSON example data, from inline examples

   |- doc  # .md documentation, from inline documentation
```

Here

- The source and working directories are optional.
- The json, examples and doc directories are populated by the sbSchemaParser

Website Files

The sbSchemaParser also generates copies of the myschema.json files into the canonical website directory, and a GH-pages version of the Markdown documentation file into the gh-pages tree processed by the GH-pages "Jekyll" processing engine. The .md file contains a permaLink directive in its YAML header, which will lead to GH-pages placing the HTML page at "https://schemablocks.org/schemas/ga4gh/myschema.html".

```
ga4gh-schemasblocks.github.io
   |
   |- schemas
   |
   |  |- ga4gh  # json version of the schema, generated from YAML
   |
   |    # => https://schemablocks.org/schemas/ga4gh/myschema.json
   |
   |  |- pages
   |
   |    |- _schemes
   |    |  |- ga4gh  # the Jekyll Markdown files for the website
```
Melanie Courtot
Michael Baudis
Isuru Liyanage
Melissa Konopko
Jonathan Fuert
Andy Yates
Ben Hutton
Bo Gao

... and many, many others in meetings & discussions
https://schemablocks.org

https://github.com/ga4gh-schemablocks/