
Phenopackets Domain Analysis

Phenopackets Domain Analysis Team

Oct 30, 2020

CONTENTS

1	Introduction	1
2	Stakeholders	3
3	Use Cases	5
3.1	Driver Use Cases	5
3.1.1	Kids First Data Resource Center (DRC): Pediatric Cancers and Structural Birth Defects . . .	6
3.1.1.1	Background and Clinical Workflow	6
3.1.1.2	FHIR Rendering of Phenotypic Information	6
3.1.1.3	Use Case	6
3.1.1.4	Technical Development and Testing	7
3.1.2	Phenotype-driven molecular genetic diagnostics	7
3.1.2.1	Background and Clinical Workflow	7
3.1.2.2	FHIR Rendering of Phenotypic Information	8
3.1.2.3	Use Case	8
3.1.2.4	Technical Development and Testing	8
3.2	Community Use Cases	9
4	Phenotypic Information	11
5	Modeling principles	13
6	Domain entities	15
6.1	Future work	15
6.1.1	Laboratory request	15
6.1.1.1	Phenopackets representation	16
6.1.1.2	FHIR representation	16
6.1.1.3	Phenopackets IG representation	16
6.1.1.4	Proposed Core IG representation	16
6.1.2	Laboratory result	16
6.1.2.1	Phenopackets representation	16
6.1.2.2	FHIR representation	16
6.1.2.3	Phenopackets IG representation	16
6.1.2.4	Proposed Core IG representation	16
6.1.3	Practitioner	16
6.1.3.1	Phenopackets representation	17
6.1.3.2	FHIR representation	17
6.1.3.3	Phenopackets IG representation	17
6.1.3.4	Proposed Core IG representation	17
6.2	Clinical findings	17
6.2.1	Phenopackets representation	17

6.2.2	FHIR representation	17
6.2.3	Phenopackets IG representation	18
6.2.4	Proposed Core IG representation	18
6.3	Clinical impression	20
6.3.1	Phenopackets representation	20
6.3.2	FHIR representation	20
6.3.3	Phenopackets IG representation	20
6.3.4	Proposed Core IG representation	20
6.4	Condition, diagnosis, etc.	20
6.4.1	Phenopackets representation	21
6.4.2	FHIR representation	21
6.4.3	Phenopackets IG representation	22
6.4.4	Proposed Core IG representation	22
6.5	Clinical documents	22
6.5.1	Phenopackets representation	23
6.5.2	FHIR representation	23
6.5.3	Phenopackets IG representation	23
6.5.4	Proposed Core IG representation	23
6.6	Encounter or visit	23
6.6.1	Phenopackets representation	23
6.6.2	FHIR representation	23
6.6.3	Phenopackets IG representation	23
6.6.4	Proposed Core IG representation	24
6.7	Identifier and identity	24
6.7.1	Phenopackets representation	24
6.7.2	FHIR representation	24
6.7.3	Phenopackets IG representation	25
6.7.4	Proposed Core IG representation	25
6.8	Individual	25
6.8.1	Phenopackets representation	25
6.8.2	FHIR representation	25
6.8.3	Phenopackets IG representation	26
6.8.4	Proposed Core IG representation	26
6.9	Karyotypic sex	26
6.9.1	Phenopackets representation	26
6.9.2	FHIR representation	26
6.9.3	Phenopackets IG representation	26
6.9.4	Proposed Core IG representation	26
6.10	Phenopacket	27
6.10.1	Phenopackets representation	27
6.10.2	FHIR representation	27
6.10.3	Phenopackets IG representation	27
6.10.4	Proposed Core IG representation	27
6.11	Sex	27
6.11.1	Phenopackets representation	28
6.11.2	FHIR representation	28
6.11.3	Phenopackets IG representation	28
6.11.4	Proposed Core IG representation	28
6.12	Specimen	28
6.12.1	Phenopackets representation	28
6.12.2	FHIR representation	28
6.12.3	Phenopackets IG representation	29
6.12.4	Proposed Core IG representation	29
6.13	Temporality	29

6.13.1	Phenopackets representation	29
6.13.1.1	string	30
6.13.1.2	Age	30
6.13.1.3	AgeRange	30
6.13.1.4	OntologyClass	30
6.13.1.5	timestamp	31
6.13.2	FHIR representation	31
6.13.2.1	string	31
6.13.2.2	instant	31
6.13.2.3	time	31
6.13.2.4	date	31
6.13.2.5	dateTime	31
6.13.2.6	Period	31
6.13.2.7	Duration	31
6.13.2.8	Age	32
6.13.2.9	Other types	32
6.13.3	Phenopackets IG representation	32
6.13.4	Proposed Core IG representation	32
6.14	Recommended workflow	32
7	Phenopackets	33
8	HL7 FHIR	35
8.1	FHIR representation	35
8.2	Phenopackets mapping	35
9	FHIR mapping	37
10	Resources	39

INTRODUCTION

This document is an effort to capture an analysis of the domain modeled in the Phenopackets schema, additional use cases and modeling requirements, how to address interoperability with the FHIR standard, and other relevant content. It is one of the artifacts developed as part of an NIH funded project.

In July of 2019 the NIH issued a notice (NOT-OD-19-122) to encourage investigators to explore the current uses of FHIR in research to ultimately improve methods for clinical researchers to use, standardize, and share electronic health data. The Broad Institute as a host institution for the Global Alliance for Genomics and Health (GA4GH) was one of two sites awarded a contract to pursue these goals under NIH/NLM contract #75N97019P00280, in partnership with the Jackson Lab, Oregon Health & Science University, Oregon State University, and Mayo Clinic. The project goals are to:

1. Coordinate with the appropriate HL7 FHIR work group(s) and develop a FHIR Implementation Guide (IG) based on the data modeled in the GA4GH Phenopackets schema,
2. Identify sets of translational research partners to demonstrate data exchange using FHIR application programming interfaces (APIs) for selected use cases,
3. Develop Driver Use Cases to inform IG development and for pilot-testing,
4. Support pilot-testing,
5. Compile pilot test results into a report,
6. Provide feedback to HL7 to revise relevant FHIR resources, profiles, and/or IGs,
7. Provide feedback to GA4GH on Phenopackets schema and application to use cases, and
8. Disseminate deliverables and obtain feedback for future development.

Note: This site is in early development. Please do not link to any specific page on this site other than the home page. Otherwise, your links will likely break as the initial content is drafted.

STAKEHOLDERS

The domain analysis will identify various stakeholders involved in developing standards and other technical solutions for the use and exchange of phenotypic information. The goal of this content is to provide a central resource for all relevant stakeholders, their working groups, significant resources provided by each stakeholder, and how to engage with each of them. The following is an initial list of stakeholders. Others will be added as part of the domain analysis and further information will be collected for each stakeholder.

- GA4GH community
 - All workstreams and driver projects related to the Phenopackets standards
 - The Clinical & Phenotypic Data Capture & Exchange [Work Stream](#)
- Phenopackets [working group](#) and [schema](#)
- HL7 FHIR community
- HL7 working groups developing relevant FHIR resources
 - Biomedical Research and Regulation [Group](#)
 - Orders and Observations [Group](#)
 - Patient Care [Group](#)
- [Kids First](#) and the [GRIN network](#)

USE CASES

This project will develop use cases to help guide the development of the technical artifacts related to the Phenopackets specification. Use cases can be proposed and provided by community members, and they will be collected as a part of this documentation for the benefit of the community and future projects.

- **Driver use cases** guide the technical development for the duration of the project. They represent a small subset of use cases that could be developed within the scientific and clinical domains that are covered by the Phenopackets specification. They are used to scope the work done by the project team in coordination with the project sponsor.
- **Community use cases** are an unconstrained set of use cases related to the Phenopackets specification. They are captured in this documentation to inform future work, including discussions regarding the scope and intent of the specification as well as the development of technical artifacts.

Please see the [driver](#) and [community](#) use cases pages for additional details.

3.1 Driver Use Cases

The following driver use cases will guide the technical development during the current project period. Please see the [community use cases](#) for additional use cases contributed by the community that could guide future work on this project.

A Driver Use Case represents the same content as a Community Use Case, but it is primarily developed by the project team for the purpose of guiding the technical development and testing activities during the project period. These use cases will be based on clinical and research topics developed in coordination with the pilot sites, involve interactions with FHIR and Phenopackets technical components, and specify the data flow, goals, and outcomes of the use case. For this project, the Driver Use Cases will be largely based on existing domain models in Phenopackets and FHIR, and existing terminologies. The primary goal of these use cases is to integrate existing technical solutions rather than evolve any specific one of them. However, in pursuing integration, specific gaps and enhancements will be needed and will be proposed for future development. These use cases will be documented and analyzed within the Domain Analysis Document.

3.1.1 Kids First Data Resource Center (DRC): Pediatric Cancers and Structural Birth Defects

This use case illustrates the need to access knowledge bases to enable research and better clinical interpretations of patient genomic data, based on observed clinical phenotypes. In this use case, a clinical researcher that is caring for a patient wants to learn more about the phenotypic and genomic profiles of other patients with similar clinical features.

3.1.1.1 Background and Clinical Workflow

The Kids First Data Resource Center (KFDRC) has supported the integration and analysis of harmonized large-scale genomic and expression data across more than 30 studies and 30,000 whole genome sequences (WGS) through the DRC's platforms. The datasets span a diverse cohort of pediatric cancers and structural birth defects that can be utilized to understand a wide range of diseases and disorders in the developmental context. The goal of the KFDRC is to make this data and information as widely available as possible for utilization in basic and translational research as well as in clinical care; for example, laboratory staff or medical geneticists could more seamlessly utilize FHIR with the appropriate tooling to retrieve the most up to date genomic landscape information relevant to the patient case at hand from KFDRC. Currently, many of these “databases” or data resources, require custom, ad-hoc or manual work for each relevant data resource to provide this functionality.

To be able to better support these capabilities, as well as have a common framework for representing this wide range of clinical data, the DRC has moved towards transitioning its current “Data Service” APIs and capabilities over to FHIR. They have piloted the technical implementation of a Kids First DRC FHIR server and are working with key stakeholders from the recently-established NIH Cloud-based Platform Interoperability (NCPI) effort to define the potential for FHIR-based platforms. Their pilot data set has been a cohort from the Pediatric Cardiac Genomics Consortium (PCGC), which has undergone whole genomic sequencing under the Kids First program and is currently loaded in our FHIR servers. Over the next month, they anticipate having several more of our studies represented in FHIR with the goal of having an initial production instance available by the end of the year for all of the released studies. For the purposes of this use case, PCGC spans a wide range of conditions and phenotypes and should be enough to evaluate the utility of utilizing FHIR to retrieve “bulk” or “cohort” data from a FHIR server based on attributes from a single patient in FHIR.

In order to add richer clinical interpretations adhering to molecular sequences and genetic variants, KFDRC has curated more than 6,000 diagnoses and 100,000 phenotypic features for the PCGC cohort. For enhanced interoperability, these conditions and phenotypes have been extensively harmonized with medical terminologies such as HPO, Mondo, and NCIt.

3.1.1.2 FHIR Rendering of Phenotypic Information

The KF FHIR implementation supports clinical information, including representations of pedigree information, conditions, phenotypic features, and biospecimens. Currently, the implementation does not support discrete genomics results, although it does include links to traditional laboratory reports.

3.1.1.3 Use Case

An infant with cardiac defects (abnormal atrial septum and pulmonary valve) was diagnosed with an atrial septal defect. As part of the diagnostic process, genomic testing was performed. No variants were found that readily explained the patient's phenotypes, but a large number of variants of unknown significance (VUS) were identified. In an effort to establish a diagnosis, the geneticist wanted to find other patients with similar phenotypic profiles and compare the genomic profiles in an attempt to identify a causative variant in a common gene. The data used in the query of the KFDRC are shown in Table 1.

The geneticist's search of the KFDRC identifies several dozen patients that have phenotypic profiles that are similar to the index case. The geneticist retrieves the genomic results for the identified patients and looks for variants in genes that are in common with the index patient's VUS profile.

An example of the type of data returned by the query to the KFDRC is shown in Table 2. Note that each table represents one patient, of which several could be returned.

3.1.1.4 Technical Development and Testing

The technical development guided by this use case will iteratively identify and profile the FHIR resources needed to represent relevant clinical data in patient-specific knowledge bases. This will include FHIR resources such as Patient, Observation, Condition, Specimen, and Genomics Report.

The technical development in support of this use case will be based on the KF FHIR implementation guide. The testing will demonstrate and evaluate the ability of the IG to represent the data included in the use case. Gaps in the IG, including the representation of genomic reports and results, will be identified and solutions will be explored; these results will be fed back to HL7 FHIR developers and to the KF team. The implementation and deployment of fixes to gaps in the specifications will not occur until later in year two of this project.

Additional details regarding the technical development and testing for this use case will be provided in the Pilot Testing Management Plan.

3.1.2 Phenotype-driven molecular genetic diagnostics

This use case addresses the need to communicate relevant clinical information from the provider's office to a laboratory and medical geneticist when an exome analysis is ordered for a patient in order to help the laboratory and medical geneticist with identifying the underlying genetic cause of the patient's clinical features.

3.1.2.1 Background and Clinical Workflow

Patients with a rare disease usually present with a set of clinical features (i.e. signs and symptoms) that in most cases do not allow a precise (etiological) diagnosis without further testing. Requesting molecular testing is not the same as requesting a routine laboratory result. Most standard laboratory tests are targeted and have well defined normal values and can be readily interpreted. The case is very different for genetic testing. Molecular testing by next-generation sequencing (e.g., exome and genome sequencing) identifies variants that vast majority of which have no clinical significance or are not related to the clinical features noticed during the initial clinical visit. Simply reporting a list of variants is not useful for a clinician, and it is not what occurs in clinical care.

The interpretation of molecular findings can be improved by comparing a patient's clinical findings with information from databases coupled with bioinformatics tools to eliminate the molecular findings that are not associated with the clinical findings and prioritize the remaining molecular findings based on how closely they can explain the clinical findings. Occasionally, it is necessary to collect additional clinical information that was not collected during the initial visit to help with selecting a candidate variant among the few top candidates.

However, there is usually a disconnect and lack of information exchange between the requesting clinician's office and the laboratory and medical geneticist offices. The clinical findings are usually not reported or are very limited. Frequently, just like other simpler lab requests, a paper form is used to request molecular testing, some clinical information might be provided in written form (hand written on the form, a printout of the visit summary), the testing is possibly further outsourced to another laboratory and it becomes very difficult or time consuming to collect further clinical information after the fact. Also, the lack of a standard and computable structure to capture and communicate these findings further complicates the usability of this information by the laboratory and medical geneticist.

3.1.2.2 FHIR Rendering of Phenotypic Information

The Phenopackets schema was developed with this use case in mind. It provides a computable representation of relevant phenotypic information to support molecular testing and analysis. It does that by modeling the relevant concepts, identifying the necessary elements for each concept (data fields), and provides recommendations for how values should be captured (i.e. terminologies or date formats)

This use case will address this clinical need and build on the knowledge gained by developing the Phenopackets schema. It will provide a set of FHIR resource profiles to enable the exchange of relevant clinical findings along with the laboratory request to provide the information needed by the medical geneticist in order to provide better and more timely clinical care for the patient.

3.1.2.3 Use Case

A female patient presents to a clinician, who observes several phenotypic features: CNS hypomyelination, intellectual disability, encephalopathy, strabismus, and scoliosis. The clinician suspects a genetic factor may be responsible and orders a clinical exome test. The phenotypic data are sent to the laboratory along with the order for the test. These data are shown in Table 1.

The laboratory uses the biospecimen to conduct the exome test, which identifies approximately 40,000 variants in the patient's genome (relative to the reference genome assembly). Even after standard filters are applied to narrow the results to variants with known or suspected pathogenicity, several hundred variants remain to be examined manually.

The laboratory uses the phenotypic information of the patient that was provided by the clinician as an additional filter, which allows the laboratory to identify a mutation in the NKX6-2 gene, which is known to be associated with the patient's phenotypes. The laboratory assembles a clinical report summarizing the findings and returns it to the clinician. The clinician diagnoses the patient with Spastic ataxia 8, autosomal recessive, with hypomyelinating leukodystrophy. The data represented in the report and the subsequent clinical note is shown in Table 2.

The information for this use case was taken from the publication PMID:28940097 (Expanding the genetic heterogeneity of intellectual disability). Exome sequencing was used to arrive at the diagnosis, Spastic ataxia 8, autosomal recessive, with hypomyelinating leukodystrophy (OMIM:617560), owing to a mutation in NKX6-2 (GRCh37, chr:10:134599256CG>C). In a typical use case, the exome or genome will have 40,000 to 5 million variants, and the analysis of the sequence data will match the clinically observed phenotypic features to a database of diseases and disease associated genes, prioritizing variants in genes that are associated with diseases with similar phenotypes.

3.1.2.4 Technical Development and Testing

The technical development guided by this use case will iteratively identify and profile the FHIR resources needed to communicate the clinical features from the point of care to a hypothetical laboratory FHIR endpoint. This will include FHIR resources such as Patient, Observation, Condition, ServiceRequest, Specimen, etc. It will also attempt to identify and profile any additional profiles necessary to communicate the laboratory molecular findings along with the initial set of clinical findings to a hypothetical medical geneticist FHIR endpoint, and eventually back to the initial clinical system that initiated the request, also through a hypothetical FHIR endpoint.

The technical development will be demonstrated by simple proof of concept tools. This will not include the building of user interfaces or modules to integrate with EHR systems. This sort of development is out of scope for this project and would come after the release of a more comprehensive Core FHIR implementation guide, which will not occur until later in year two of this project.

Additional details regarding the technical development and testing for this use case will be provided in the Pilot Testing Management Plan.

3.2 Community Use Cases

A Community Use Case describes the specific information needs, data types, and any relevant domain or technical aspects that a stakeholder needs to support their goals. The community use cases, as opposed to the *Driver Use Cases*, will primarily be developed by the wider community, not the project team, and will be retained for future use by the Phenopackets project. The project team will establish a workflow for how to collect and evolve these use cases for the benefit of all stakeholders. These use cases will initially be captured as a GitHub issue in the [domain analysis repository](#) to facilitate community discussion and development, and will be merged into this document when mature.

PHENOTYPIC INFORMATION

This section will focus on capturing and analyzing examples of phenotypic information independent of any existing data models. This content will also be an input into the process of identifying and defining important domain entities.

MODELING PRINCIPLES

DOMAIN ENTITIES

The important aspects of domain analysis include identification of relevant domain concepts, the documentation of how people in the domain understand and use those concepts, and what data is collected about them or in relation to them. Specific examples of domain entities are the concepts of person, patient, study subject, diagnosis, signs and symptoms, family, and pedigree. People in the domain have a clear and intuitive understanding of these concepts but they also appropriately adjust their understanding and use of these concepts based on the domain context. This frequently leads to difficulties when it comes to data modeling and the often implicit shift of semantic meaning inhibits interoperability.

This section aims to identify these concepts and represent them as modeling entities. This process usually involves capturing the various ways these concepts exist in the domain, giving clear definitions and making clear distinctions, and creating new entities with new labels when a domain concept might be too general or vague and can imply very different entities for different people in the domain or in different domain contexts.

Although the analysis in the following pages should be based on the domain and how domain experts understand it rather than how it's modeled in any existing information model, these pages will likely refer to existing representations as a way to identify aspects of the domain that the model intended to represent. Models are usually developed for specific use cases and examining their approach will likely lead us to identifying aspects of the domain that we might not be aware of. However, we are not constrained by the specific definition and modeling decisions from those models. Also, a common modeling error is the “over fitting” of such models to the specific use case or project at hand. This over fitting leads to difficulties with reusing these models for other projects and eventually to the proliferation of over fitted models. This is a major interoperability barrier. In the following pages those models are simply looked at as a tool to identify the underlying domain they are intending to represent and provide one or more alternative representations within the FHIR framework.

*Any workflow related comments can be [posted here](#)

6.1 Future work

The following entities are deferred to later milestones.

6.1.1 Laboratory request

Sending a Laboratory request for genetic analysis.

GitHub issue: [ISSUE](#)

6.1.1.1 Phenopackets representation

N/A. This concept is not represented by Phenopackets, but it is part of the driver Use Cases for this project.

6.1.1.2 FHIR representation

FHIR has a [ServiceRequest](#)

6.1.1.3 Phenopackets IG representation

N/A

6.1.1.4 Proposed Core IG representation

We will profile [ServiceRequest](#).

6.1.2 Laboratory result

Define entity...

GitHub issue: [ISSUE](#)

6.1.2.1 Phenopackets representation

Discuss...

6.1.2.2 FHIR representation

Discuss...

6.1.2.3 Phenopackets IG representation

Discuss...

6.1.2.4 Proposed Core IG representation

Discuss...

6.1.3 Practitioner

Define entity...

GitHub issue: [ISSUE](#)

6.1.3.1 Phenopackets representation

Discuss...

6.1.3.2 FHIR representation

Discuss...

6.1.3.3 Phenopackets IG representation

Discuss...

6.1.3.4 Proposed Core IG representation

Discuss...

6.2 Clinical findings

This represents any type of observation of clinically relevant things or situations. However, the analysis in this page is primarily focused on:

- phenotypic features

GitHub issue: [ISSUE](#)

6.2.1 Phenopackets representation

The Phenopackets schema represents phenotypic features with the [PhenotypicFeature](#) entity. It captures various aspects of a clinical characteristic of a subject, usually an abnormality. It also defines a few fields to capture different types of qualifiers including negation, severity, and phenotype-specific qualifiers. There is also a representation of the age of onset of the phenotypic feature.

For our current exome analysis use case we are primarily concerned with capturing the type of a phenotype and a positive vs negative assertion. Other aspects of the Phenopackets PhenotypicFeature will be addressed based on future driver use cases.

It appears that the .severity field's recommended values are a subset of the .modifier field's recommended values.

6.2.2 FHIR representation

This will be a FHIR [Observation](#). It captures phenotypes and other sorts of observations.

A FHIR Observation, in its most general sense, is a key/value pair with additional optional key/value components to capture parts of the overall observation.

The use of this very flexible key/value model can lead to several patterns and this complicates interoperability. The FHIR specification recognizes this and give guidance as [described here](#).

The language regarding the [use of Observation components](#) is also relevant to our usecase(s).

6.2.3 Phenopackets IG representation

It uses FHIR Observation however it models the various types of PhenotypicFeature qualifiers with Extension profiles instead of using the Observation components approach, as [shown here](#)

6.2.4 Proposed Core IG representation

We will:

- Represent PhenotypicFeature with Observation. One to one instance.
- Represent PhenotypicFeature.type as .code and follow the item 2 in the [FHIR guidance here](#), where .code captures the clinical finding (phenotypic feature, in this case)
- Represent PhenotypicFeature.severity and .modifier as Observation.component entries as justified by [this language](#) as opposed to needing Extension profiles.
- Possibly represent PhenotypicFeature.onset as a component. The Observation examples page has an example of an observation that has time built into (pre-coordinated with) the code (the Apgar score). We can argue that our PhenotypicFeature are time dependent but we don't have onset precomposed into our coding system. However, a PhenotypicFeature will post compose the onset indexed phenotype with Observation components. We believe there are several value[x] available to accommodate the representation of onset.

Given that the PhenotypicFeature.severity is a subset of codes for .modifier, we can represent this with a single modifier component profile that can be instantiated as many times as needed to add the needed qualifiers. The value set for .code will be a small set of very different types of qualifiers, and the value set for .valueCodeableConcept will be the subtree of those codes.

The PhenotypicFeature [example here](#) can be represented with this Observation instance but other minor variations (especially for the use of .code and .value[x]) are possible.

```
{
  "resource": {
    "resourceType": "Observation",
    "code": {
      "coding": [
        {
          "code": "HP:0000520",
          "system": "http://HP",
          "display": "Proptosis"
        }
      ]
    },
    "valueBoolean": true,
    "component": [
      {
        "code": {
          "coding": [
            {
              "code": "HP:0012824",
              "display": "Severity",
              "system": "http://HP"
            }
          ]
        }
      ]
    ],
    "valueCodeableConcept": {
      "coding": [
        {

```

(continues on next page)

(continued from previous page)

```

        "code": "HP:0012825",
        "display": "Mild",
        "system": "http://HP"
      }
    ]
  },
  {
    "code": {
      "coding": [
        {
          "code": "HP:0012823",
          "display": "Clinical modifier",
          "system": "http://HP"
        }
      ]
    },
    "valueCodeableConcept": {
      "coding": [
        {
          "code": "SomeAppropriateModifier",
          "display": "Some appropriate modifier",
          "system": "http://HP"
        }
      ]
    }
  },
  {
    "code": {
      "coding": [
        {
          "code": "HP:0003674",
          "display": "Onset",
          "system": "http://HP"
        }
      ]
    },
    "valueCodeableConcept": {
      "coding": [
        {
          "code": "HP:0003577",
          "display": "Congenital onset",
          "system": "http://HP"
        }
      ]
    }
  }
]
}

```

6.3 Clinical impression

This is an entity related to a high level clinical assessment and workflow/summary. It is distinct from but closely related to the more granular observations, diagnoses, investigations entities.

FHIR does a fairly good job defining and modeling this entity with the [ClinicalImpression](#) resource.

The FHIR ClinicalImpression model is good knowledge to keep in mind while we go through the analysis and modeling of the smaller supporting entities. However, a full analysis and modeling of this domain entity is premature at this point, and not clearly needed by year 1 use cases, so it will be deferred until it is clearly needed and we have resolved and implemented most of the other supporting entities needed for this entity.

There are aspects of the Phenopackets schema modeling that can be refactored into the FHIR ClinicalImpression resource, or the other FHIR related resources referenced by ClinicalImpression. This could be a year 2 analysis topic, or it can also be left to downstream IGs to model along with any reporting or document modeling. The Core IG can focus on providing a well defined set of building blocks for those downstream IGs.

GitHub issue: [ISSUE](#)

6.3.1 Phenopackets representation

Discuss...

6.3.2 FHIR representation

Discuss...

6.3.3 Phenopackets IG representation

Discuss...

6.3.4 Proposed Core IG representation

Discuss...

6.4 Condition, diagnosis, etc.

This entity represents the usual sense of a clinical diagnosis, a patient having a condition that requires medical care, etc. The line between a phenotype and a diagnosis is not always clear. However, FHIR does provide some guidance and this entity is meant to be aligned with FHIR's definition of the [Condition](#) resource.

The first two sections of the FHIR documentation for Condition ([scope](#) and [boundaries](#)) provides a good overview of Condition vs other similar FHIR resources and we will adopt that analysis here.

In addition to the FHIR language to clarify condition vs phenotype, **a condition should be considered as a conclusion or inference that is based on some other information or observations**, usually simpler or more granular. Practitioners in the domain are intuitively aware these distinctions and act accordingly even if our written definitions give the impression that there is no clear boundaries between a phenotype and a condition. For example, if we only have a fact that there is a low body temperature, is this a medical condition that needs attention or is it just a simple phenotype of a body? In clinical practice it is a medical emergency that needs immediate treatment but during an autopsy it is just a characteristic of the body that needs no intervention yet that fact can still be used to make other

inferences that are still clinically relevant (such as time of death). So, in this case, the first scenario the low body temperature is a hypothermia condition but in the latter case it would be inappropriate to make that inference. The hypothermia condition, in addition to needing to have a phenotype of low body temperature, also implies that there is an expected body temperature, that the patient is alive, and that the assertion of hypothermia is made by a qualified practitioner as opposed to an informal inference by any other type of observer.

Also, this page is not primarily focused on documents, reporting, etc. It is focused on the underlying meaning and structure of the various entities referenced in such reports and documents.

GitHub issue: [ISSUE](#)

6.4.1 Phenopackets representation

This concept is represented in Phenopackets by [the Disease entity](#). It is focused on capturing the coded type of the condition, onset in the form of an age-related quantity or a coded value for an onset category, and staging information that is captured in two dedicated fields. The “disease_stage” appears to be general and can be used for any disease as long as there is some coded staging for it, and a “tnm_finding” that is specifically for tumor-related staging. The Phenopackets documentation gives further guidance for these fields, how they should be used, and suggest a subset of NCIT codes for each field.

In the Phenopackets [schema diagram](#) and in [the Interpretation top level element](#) there is a sense of a higher level interpretation that is in addition to and separate from the Disease modeling. This seems to be dedicated to a genomics analysis interpretation, not a general overall interpretation of what is in a phenopackets instance.

The Interpretation model also nests [a Diagnosis entity](#) that appears to capture a relationship between a Disease and a [GenomicInterpretation](#), which is another nested Entity within Interpretation. There is also a [GenomicInterpretation Status enumeration](#) within Interpretation.

6.4.2 FHIR representation

The basic sense of a condition or diagnosis is represented with [the Condition resource](#). FHIR Condition provides modeling to capture all of the issues described above for Phenopackets Disease. It also has a .staging complex structure that allows for multiple yet distinct entries for staging related information. It is also able to capture severity, related body sites, etc. It also has [a .onset\[x\] field](#) that allows for Age (i.e. Quantity) and Range values (in addition to few other value types) to specify the onset. However, it does not allow for a coded type of onset, i.e. no onsetCodableConcept, so an extension may be necessary.

Further discussion of Phenopackets modeling of Disease, Diagnosis, Interpretation is needed but with a superficial understanding of those Phenopackets entities, it appears that some aspects of those Phenopackets entities can be subsumed by FHIR Condition or ClinicalImpression. There is also the DiagnosticReport resource if a more formal document like model with an author is needed.

It appears that we will need to perform some significant amount of analysis here that will clarify and align the relevant parts of Phenopackets and FHIR to cover this aspect of the domain. It is also likely that we will be able to generalize the Phenopackets representation in a way that will make it easier to reuse the idea of an Interpretation, Diagnosis, etc. in a way that is not so closely tied or constrained to genomics. :TODO:move this language to some other part of this page.

6.4.3 Phenopackets IG representation

It models Phenopackets Disease with a FHIR Condition as [can be seen here](#). It also provides two extensions to capture coded onset and tumor staging, i.e. the Phenopackets Disease.tnm_finding field.

It models Phenopackets Interpretation, [as seen here](#), as a GenomicsReport from the Genomics Reporting IG, which in turn is based on FHIR DiagnosticReport.

We still need to explore how this IG captures the other Phenopackets Interpretation related modeling described above.
:TODO:

6.4.4 Proposed Core IG representation

We will model the disease/condition part of the above analysis with a FHIR Condition. Also:

- All staging related information will be modeled with the built in Condition.stage complex structure.
- Onset will be captured with Condition.onset[x] - If a coded onset is needed we may use an Observation over the Condition instance to capture that information. In this case, it is likely that the Observation can be used as Condition.evidence. For example, for a type 1 diabetes condition instance, it would be possible to have an Observation of a coded onset value and have that Observation be a Condition.evidence for the diagnosis of type 1 diabetes. We'll have to see if this can not be done consistently. Alternatively, an extension could be developed for onset that supports CodeableConcepts.
- We will need to develop ValueSet resources to represent or subsume the Phenopackets Disease recommendation for terminology bindings.

The other aspects of Phenopackets described in this page are pending further analysis but the FHIR Condition resource is a basic building block towards capturing the overall Phenopackets modeling of this part of the domain. The ClinicalImpression resource is another closely related model that needs further analysis along with the Phenopackets modeling.

A tentative thought is that we should stay away from the idea of capturing information in this Core IG in FHIR resources such as **document**. Other downstream IGs can implement this use case, or we can prototype few types of documents in this Core IG if needed. However, basic facts, relationship, information objects, etc. should not be directly or primarily be captured in a document-like FHIR resource. The Core IG should be focused on A FHIR information graph rather than also embedding parts of that graph in documents.

6.5 Clinical documents

This represents any type of document or file that is clinically relevant. However, for this project we're primarily concerned with the following types of documents:

- A phenopacket message, object, document, etc. in its native format.
- Other...

GitHub issue: [ISSUE](#)

6.5.1 Phenopackets representation

Phenopackets schema has a top level entity that represents the container for the information in a phenopacket instance.

6.5.2 FHIR representation

FHIR allows various resources to reference a document through a [DocumentReference](#). This resource can be used to capture a reference to a phenopacket document in the context of a FHIR API. We'll need to determine the:

- [DocumentReference.type](#)
- [DocumentReference.content.format](#)
- [DocumentReference.content.attachment.contentType](#)

The content of a phenopacket can be base64-encoded and added to [DocumentReference.content.attachment.data](#)

6.5.3 Phenopackets IG representation

N/A

6.5.4 Proposed Core IG representation

Use a [DocumentReference](#) and inline a phenopacket while submitting a laboratory request for UC 2.

6.6 Encounter or visit

An individual encounter or visit for an individual to obtain related services. It is the context for related information.

GitHub issue: [ISSUE](#)

6.6.1 Phenopackets representation

This concept is not represented in the Phenopackets specification, but it is required to support Use Case 2 (UC2).

6.6.2 FHIR representation

FHIR provides an [Encounter](#) resource.

6.6.3 Phenopackets IG representation

N/A

6.6.4 Proposed Core IG representation

We will use the encounter resource and related ones to organize the initial clinical information that led the practitioner to order the laboratory request for UC2. It will be the needed context for what should be extracted into a phenopacket instance that represents the phenotypic information noted during that encounter.

6.7 Identifier and identity

Identifiers are a critical part of linking an information system to entities in the real world that the information system is intended to capture. Identifiers are also needed for the information system itself to be able to link the various information objects with each other to maintain the information graph. These two types of identification are very different. FHIR has a good discussion of this issue [here](#) and [here](#)

GitHub issue: [ISSUE](#)

6.7.1 Phenopackets representation

Phenopackets describes identifiers in several places, including [here](#), [here](#), [here](#), and [here](#). Identifiers are used in Phenopackets to define object identity and as cross-references defined by external systems. Phenopackets also uses identifiers as foreign keys to other parts of a Phenopackets message, maintaining links between objects, rather than as true domain business identifiers. Phenopackets also defines the [alternate_ids field](#), appears to be more of a business identifier, or is able to accommodate that function to some degree.

Within the PP Individual structure, the id field is defined as “the primary identifier for the individual and SHOULD be used in other parts of a message when referring to this individual - for example in a Pedigree or Biosample. The contents of the element are context dependent, and will be determined by the application.”

The type of the identifier attribute is “string”, with no limitations on length or character set.

6.7.2 FHIR representation

FHIR models logical or resource instance identity with the “id” field. The value of this id field is a simple type, i.e. a [simple string value with some limitations](#).

Identifiers for business identifiers are represented with the [Identifier data type](#) and usually the “identifier” field holds instances of that data type. The identifier field is used on various resources that are likely to have a business identity, where the “system” attribute . It is not available on all resources the way the “id” field is. The Identifier data type captures not only the value of an identifier, but also the system or namespace that assigned the identifier and its type and intended use. Therefore, this datatype is an elegant approach to encapsulating relevant metadata related to an identifier so that a consuming system can computationally determine how it might be used.

The type of the Identifier.value field is “string”, but FHIR restricts the value of strings by length (1024x1024 characters) and content (Unicode, with restrictions).

This is similar to the [FHIR logical id field](#) and it also matches the simple string based representation of this value

6.7.3 Phenopackets IG representation

The IG doesn't clearly describe how the PP are mapped. However, it does add a "must support" to the FHIR identifier field.

6.7.4 Proposed Core IG representation

The PP Individual.id attribute is mappable to the FHIR id type. However, because PP places heavy emphasis on the context and business use of this value, the Identifier type (which captures system, type, and use) is more appropriate. The Identifier type would also better support the use of multiple identifiers for an Individual, and its use is more consistent with FHIR recommendations. Furthermore, the use of Identifier allows for preserving object identities when objects are moved between systems and the message-specific "logical id" is lost. The language supporting this is [here](#).

The alternative identifier has to be captured in the FHIR identifier field whenever possible to be consistent with identity representation in FHIR.

Giving a set of alternative identifiers in some phenopackets instance, the identifiers could clarify which FHIR resources to instantiate for a PP Individual. For example, if one of those alternate identifiers is a medical record number, this clearly implies some FHIR Patient aspect to the phenopacket. However, if one of the identifiers is a SSN, or something like that, this implies a FHIR Person aspect to the PP instance. An identifier that is not clearly related to a Patient role/context should canonically be mapped to a Person instance. However, it is also possible to duplicate the representation of this identifier on the Patient instance as well if and when needed. FHIR provides guidelines for how to update information between the different FHIR resources that represent people in the world and their role specific instances, [here](#), [here](#), etc.

6.8 Individual

This is an individual person in the domain independent of any role in any clinical or research setting. It subsumes the patient role.

GitHub issue: [ISSUE](#)

6.8.1 Phenopackets representation

The Phenopackets specification uses [Individual](#) to represent a patient (the proband of the Phenopacket), but similar concepts within the specification are used to represent people in other ways. The model for this element is not focused on capturing encounters, providers, etc. and there is more focus on person-to-person relationships to aid interpretation of genetic data. [This language](#) supports this interpretation and the attributes also support this structure.

Phenopackets captures both sex and karyotypic sex in this entity.

6.8.2 FHIR representation

FHIR represents the concept of individuals in many ways. The core Patient resource entity represents the role of a person as a patient in a healthcare organization, rather than representing an individual. The language that supports this interpretation is [here](#) and [here](#), and in few other places in the Patient documentation page.

FHIR also has a resource that represents [Person](#), which is meant to be a role-neutral representation of an individual. See also [RelatedPerson](#), although this resource is less relevant to the current driver use cases. Also see [this section](#).

The FHIR Patient documentation provides good guidance for issues related to sex, gender, and clinically based sex and gender facts. See [this section](#).

6.8.3 Phenopackets IG representation

This IG maps a PP Individual to a [FHIR Patient](#)

6.8.4 Proposed Core IG representation

A Phenopacket “Individual” should be primarily mapped to a FHIR Person as it is the closest semantic match. If there is a need to also capture the Phenopacket’s instance in the context of a FHIR Patient (because information is present that is related to a specific Patient instance in a clinical setting), then a FHIR Patient instance should also be created for this purpose and linked to the Person instance as the primary representation of existence of the Individual in the world. Also note that FHIR’s main “integration” point for most or all FHIR instances is the Patient resource (not Person) so it is almost a requirement to instantiate both Person and Patient to be able to capture other information with the existing FHIR resources. We should follow the guidelines described in the Patient and Person FHIR documentation pages and instantiate one or both as needed based on what other information we need to attach to them. See further discussion below.

6.9 Karyotypic sex

Karyotypic sex (KS) is the determination of sex chromosomes in an individual.

GitHub issue: [ISSUE](#)

6.9.1 Phenopackets representation

Phenopackets captures KS as an [attribute on Individual](#) and provides a [value set for permissible values](#).

6.9.2 FHIR representation

FHIR does not have a dedicated named field for this domain element. However, it provides guidance for this and closely related domain concepts [here](#) and suggests an approach for representing this concept using observations, since there are multiple types of clinical sex (e.g., karyotypic, gonadal, ductal, phenotypic) that could be captured as coded observations.

6.9.3 Phenopackets IG representation

The IG uses a FHIR extension to capture KS, as shown [here](#).

6.9.4 Proposed Core IG representation

The PP karyotypic sex is mapped to an Observation instance. The PP value set is represented with a FHIR valueset that builds on existing FHIR coding systems (assuming HPO will be one of them), or a proposed new coding system or new content for a new coding system if necessary.

Any binding declared in Observation profiles for this element should be preferred or extensible at most, not required. [See here](#) As it says at the end of that section, the binding can be further constrained for project-specific purposes but for wider interoperability purposes it is very harmful to have a required binding on a very specific value set, as this forces a profile to be “over fitted” to a specific use case and the profile will not be reusable by others, thereby limiting the interoperability value of the profile.

6.10 Phenopacket

This page is for the analysis of a phenopacket. The term *phenopacket* here means an instance of the Phenopackets graph shown [here](#) and further defined [here](#)

What is the context of a phenopacket? What does it mean to extract or submit a phenopacket from/to an EHR? Everything in an EHR is usually linked to some provider, on a specific date, during an encounter, asserted by a provider, etc. All this context is missing from the Phenopackets schema. How can this context be made up for or ignored in an EHR? Intuitively, it's not likely to be able to *enter* a phenopacket into an EHR without it being attested to by a provider, during an encounter, etc. Also, the various components of a phenopacket as defined by the Phenopackets schema are likely to be instantiated multiple times, even in duplicate, in an EHR during various encounters for the same patient and same clinical conditions. Also, the set of instantiated parts of the phenopacket schema might vary from one encounter to another. For example, provider x on date y might say a patient has phenopacket instance p1 but on another date have a different combination of instances to form phenopacket p2. How will all this be mappable to the simpler use of Phenopackets outside the context of EHR, providers, visits, purpose, etc.?

GitHub issue: [ISSUE](#)

6.10.1 Phenopackets representation

Discuss...

6.10.2 FHIR representation

There are multiple options to group FHIR resources in FHIR. Please be familiar with the nitty gritty details of [Bundle](#) and [Composition](#), and to a lesser degree with [Linkage](#), [List](#), before discussing FHIR options for representing Phenopackets' meaning of a phenopacket. There are few other relevant FHIR resources but the above are a good starting point.

6.10.3 Phenopackets IG representation

This IG represents a phenopacket as FHIR Composition but this might not be the best choice based on the spec language for that resource. Specific justification with citations for not doing this will be added here shortly. A FHIR Bundle might be a more appropriate container but this is pending on clarifying what a phenopacket instances should stand for inside and outside an EHR, and have some of the questions at the beginning of this page answered.

6.10.4 Proposed Core IG representation

Discuss...

6.11 Sex

PP Definition: Observed (phenotypic) apparent sex of the individual.

GitHub issue: [ISSUE](#)

6.11.1 Phenopackets representation

Phenopackets captures Sex as an attribute on Individual and provides a value set for permissible values. The description specifies this is an observation rather than a self-reported value.

6.11.2 FHIR representation

FHIR provides a field for gender, with an associated value set for Administrative Gender. It should be noted that the concept of sex is distinct from that of gender, as the former is physiological and the latter describes self-identified behaviors and roles. FHIR does not support the concept of sex in FHIR 4.0.1.

6.11.3 Phenopackets IG representation

The IG appears to map the concept of Sex to FHIR Patient.gender. FHIR has the same values in the required ValueSet as Phenopackets has.

6.11.4 Proposed Core IG representation

FHIR suggests either an extension, as provided by US Core, or an Observation instance for capturing this apparent sex in a clinical setting as opposed to an administrative settings. See [this language](#) The extension approach is limited compared to an observation asserted by a provider or a Patient in a specific context. Capturing it as an extension is straightforward if there are no doubts clinically but for situations where the assignment is difficult, or later medical care is necessary to determine it, Observation instances become more appropriate. It is also possible to represent it both ways. The extension on Patient holds the same value as the latest Observation of this feature.

6.12 Specimen

Both driver use cases have a need to represent a specimen of type blood, the specific analyte, and possibly a tissue type or specimen type qualifier.

GitHub issue: [ISSUE](#)

6.12.1 Phenopackets representation

Phenopackets models this with [the Biosample entity](#). Its definition is focused on molecular analysis of an extracted analyte/substrate.

6.12.2 FHIR representation

FHIR models this with [the Specimen resource](#).

6.12.3 Phenopackets IG representation

Creates a [FHIR Specimen profile](#) and adds several extensions to capture related clinical facts (as modeled in Phenopackets) as extensions on Specimen.

6.12.4 Proposed Core IG representation

This analysis is still incomplete but it is likely that this can be captured with a combination of FHIR Specimen, along with references to related Condition, Observation, etc., to refactor the various aspects of the Phenopackets Biosample entity instead of needing the many custom extensions. For our driver use cases for year 1 we will not need these clinical aspects of Phenopackets Biosample so the mapping in this case will only involve profiling Observation. Other aspects will be dealt with as they arise in use cases.

6.13 Temporality

This page discusses the representation of all concepts related to time, including discrete time points, periods, quantities, and codes that represent any of those.

GitHub issue: [ISSUE](#)

6.13.1 Phenopackets representation

The Phenopackets specification refers to time-based data elements in many places and represents time data in many different ways, including:

Class	Attribute	Datatype(s)
Age	age	string
AgeRange	start	Age
AgeRange	end	Age
Disease	onset	Age AgeRange OntologyClass
Individual	date_of_birth	timestamp
Individual	age	Age AgeRange
MetaData	created	timestamp
PhenotypicFeature	onset	OntologyClass
Update	timestamp	timestamp

Some elements of the Phenopackets specification capture time as a primitive datatype (string, timestamp) with additional formatting constraints. Other elements capture time in complex datatype objects (Age, AgeRange, OntologyClass) and may be polymorphic (permitting any one of several types to be used). Due to this heterogeneity within the specification, we must analyze each type individually.

6.13.1.1 string

Used by: Age.age

This string conforms to the format specified by [ISO8601](#) Duration:

Durations define the amount of intervening time in a time interval and are represented by the format P[n]Y[n]M[n]DT[n]H[n]M[n]S or P[n]W. In these representations, the [n] is replaced by the value for each of the date and time elements that follow the [n]. Leading zeros are not required, but the maximum number of digits for each element should be agreed to by the communicating parties. The capital letters P, Y, M, W, D, T, H, M, and S are designators for each of the date and time elements and are not replaced.

- P is the duration designator (for period) placed at the start of the duration representation.
 - Y is the year designator that follows the value for the number of years.
 - M is the month designator that follows the value for the number of months.
 - W is the week designator that follows the value for the number of weeks.
 - D is the day designator that follows the value for the number of days.
- T is the time designator that precedes the time components of the representation.
 - H is the hour designator that follows the value for the number of hours.
 - M is the minute designator that follows the value for the number of minutes.
 - S is the second designator that follows the value for the number of seconds.

6.13.1.2 Age

Used by: AgeRange.start, AgeRange.end, Disease.onset, Individual.age

This complex datatype has a single attribute (age), which is a formatted string (see above).

6.13.1.3 AgeRange

Used by: Disease.onset, Individual.age

This complex datatype has two attributes (start and end), which are of type Age (see above).

6.13.1.4 OntologyClass

Used by: Disease.onset, PhenotypicFeature.onset

This complex datatype is used to capture concept codes. It is comprised of two required attributes: an id in [CURIE](#) format (where the *prefix* portion of the CURIE is mapped to a Resource through the [MetaData class](#) and a label that represents the name of the concept from the Resource. OntologyClass could be used to represent coded periods of time.

6.13.1.5 timestamp

Used by: Individual.date_of_birth, MetaData.created, Update.timestamp

The timestamp type is a string that conforms to the format specified by [ISO8601](#) with time zone support (UTC).

6.13.2 FHIR representation

FHIR supports a number of datatypes that support capturing dates and times, which are summarized below.

6.13.2.1 string

The primitive [string](#) is a sequence of Unicode characters that shall not exceed 1024*1024 characters in length. There are additional restrictions regarding whitespace.

6.13.2.2 instant

The primitive [instant](#) represents a timestamp in YYYY-MM-DDThh:mm:ss.sss+zz:zz format that has a precision of at least seconds and includes a time zone.

6.13.2.3 time

The primitive [time](#) is a string in hh:mm:ss format. Additional minor constraints are specified.

6.13.2.4 date

The primitive [date](#) is a string in YYYY, YYYY-MM, or YYYY-MM-DD format, without time zone.

6.13.2.5 dateTime

The primitive [dateTime](#) is a string in YYYY, YYYY-MM, YYYY-MM-DD or YYYY-MM-DDThh:mm:ss+zz:zz format. If hours and minutes are present then a time zone must be specified. Additional minor constraints are specified.

6.13.2.6 Period

The complex type [Period](#) specifies a range of times that is defined by a start and end time, both of the [dateTime](#) type.

6.13.2.7 Duration

The complex type [Duration](#) is constrained from [Quantity](#). It captures a value (type: decimal) and a code that specifies a unit (of time). If the system that defines the code is present, it must be UCUM.

6.13.2.8 Age

The complex type [Age](#) is constrained from [Quantity](#). It captures a value (type: decimal) and a code that specifies a unit (of time) that is appropriate for Age. If the system that defines the code is present, it must be UCUM.

6.13.2.9 Other types

FHIR supports other datatypes that could represent time, which are not discussed here. CodeableConcept and Coding are discussed elsewhere, generically. SimpleQuantity and Quantity are less appropriate than the alternative specialized types described above. Timing describes the occurrence of an event that may occur multiple times and is not relevant to Phenopackets.

6.13.3 Phenopackets IG representation

Discuss....

6.13.4 Proposed Core IG representation

Discuss...

*Add a page to the */domain-entities* folder and it will appear in the above list.

6.14 Recommended workflow

1. Identify an in scope domain entity.
2. Create a page for it by copying the */domain-entities/_template.rst.off* file and naming it following the pattern used by the other files in that same directory.
3. Fill in or replace the template's content
4. Create a GitHub issue for the domain entity to support ongoing discussion of the entity#. When appropriate, add any needed mapping information according to the mapping workflow (yet to be defined) and build any needed FHIR conformance resources to implement the FHIR representations for the domain element. This is done under the [Core IG repository](#)
5. Any newly developed content for the DAD or the Core IG should be submitted with a GitHub PR for review before merging into the GH master branch.

As part of this workflow, and to facilitate the team's ability to review source material, annotate it, have quick informal comments or questions about it, and to be able to do all this within the context of the original source material, we are copying the source material into a set of Google Docs pages in a folder dedicated for this purpose. As you read through the analysis of the domain entities listed below, you will find various links to comments in these copies as a way to identify and communicate relevant portions of the source content. For example, the Phenopackets Individual and FHIR Patient and Person resource documentation is copied to individual pages and various parts of this content is sited in the analysis of those domain entities. When following any of these links, please be patient until the linked page is fully loaded and the page is scrolled to the specified linked area or comment.

The folder can be [found here](#) and any comments are viewable by all. Each page has a link at the top to the original source of the copied content. Join the project's team to be able to comment. However, anyone with a GitHub account should be able to comment on the corresponding issue if they don't have access to these docs as an alternative way for providing feedback.

PHENOPACKETS

Phenopackets is a well established GA4GH schema for capturing and exchanging structured phenotypic data to support common research and clinical workflows. In order to achieve the goals of this project, the current Phenopackets model will be examined, including the underlying domain it represents and how technical data is represented, which will enable a rigorous comparison to the FHIR standard. Careful analysis of those two specifications, with specific consideration of their semantic models and data representations, will identify new areas of development. We anticipate that the outcome of this process will be the enhancement of both standards through the delivery of technical guidance and tools to support data integration and interoperability.

The [Phenopackets project](#) describes it as:

The Phenopacket Schema represents an open standard for sharing disease and phenotype information to improve our ability to understand, diagnose, and treat both rare and common diseases. A Phenopacket links detailed phenotype descriptions with disease, patient, and genetic information, enabling clinicians, biologists, and disease and drug researchers to build more complete models of disease (see Disease for the distinction between disease and phenotypic feature). The standard is designed to encourage wide adoption and synergy between the people, organizations and systems that comprise the joint effort to address human disease and biological understanding.

HL7 FHIR

The HL7 FHIR specification was mostly developed for clinical systems and is heavily focused on established clinical workflows. It is not as well developed for the research domain, or for capturing more granular clinical facts such as phenotypic level information. However, FHIR does have some foundational building blocks that could be extended (with FHIR extensions) or could be further developed to accommodate both of these areas. The analysis of FHIR to represent detailed clinical phenotypes and related data will be focused on two main areas.

8.1 FHIR representation

The analysis will examine and document how the FHIR framework represents clinical phenotypic information and related data, including which FHIR resources are most applicable to the representation of those data. Particular attention will be paid to identifying gaps in the FHIR specification that need to be addressed to better support our use cases.

8.2 Phenopackets mapping

This part of the FHIR analysis will specify in detail how the Phenopackets schema maps to FHIR resources. This process will provide the mappings needed for technical development and implementation of a FHIR IG, but it will also help identify possible enhancements to the FHIR and Phenopackets specifications to bring them into better alignment in how they represent the underlying domain entities.

FHIR MAPPING

The FHIR mapping workflow is focused on taking the analysis and recommendations from the *domain entities* and converting it to exact mappings to FHIR resources, elements, data types, values, etc.

The workflow is primarily based on Google Sheets, GitHub issues, and prototyping the chosen FHIR implementation in the Core IG.

- The [Google Sheet](#)
 - The set of green columns capture domain related information regardless of its current representation in Pheopackets or FHIR
 - The next set of blue columns capture Phenopackets' representation.
 - The next set of yellow columns capture FHIR's representation.
 - A row represent a mapping from the domain to Phenopackets and/or FHIR
 - Additional columns will be used for other workflow aspects such as status, GH issue, review comments, etc.
- GitHub issues at the [Core IG issue tracker](#)
- GitHub pull requests (PRs) for changes to the Core IG. Each pull request is [built and published](#) with the FHIR publisher tool. Links to these PR specific builds can be found at that page.

RESOURCES

This page will contain a listing of resources and tools that are relevant to the phenotypic domain.