Ontological Modelling and Reasoning of Phenotypes

Alexandr UCITELI[a,1] , Christoph BEGER[a,b], Toralf KIRSTEN[c], Frank A. MEINEKE[a] and Heinrich HERRE[a]

[a] *Institute for Medical Informatics, Statistics and Epidemiology (IMISE), University of Leipzig, Germany*

[b] *Growth Network CrescNet, University of Leipzig, Germany*

[c] *Faculty of Applied Computer and Biological Sciences, University of Applied Sciences Mittweida, Germany*

Abstract. The successful determination and analysis of phenotypes plays a key role in the diagnostic process, the evaluation of risk factors and the recruitment of participants for clinical and epidemiological studies. The development of computable phenotype algorithms to solve these tasks is a challenging problem, caused by various reasons. Firstly, the term 'phenotype' has no generally agreed definition and its meaning depends on context. Secondly, the phenotypes are most commonly specified as non-computable descriptive documents. Recent attempts have shown that ontologies are a suitable way to handle phenotypes and that they can support clinical research and decision making.

The SMITH Consortium is dedicated to rapidly establish an integrative medical informatics framework to provide physicians with the best available data and knowledge and enable innovative use of healthcare data for research and treatment optimization. In the context of a methodological use case "phenotype pipeline" (PheP), a technology to automatically generate phenotype classifications and annotations based on electronic health records (EHR) is developed. A large series of phenotype algorithms will be implemented. This implies that for each algorithm a classification scheme and its input variables have to be defined. Furthermore, a phenotype engine is required to evaluate and execute developed algorithms.

In this article we present a Core Ontology of Phenotypes (COP) and a software Phenotype Manager (PhenoMan), which implements a novel ontology-based method to model and calculate phenotypes. Our solution includes an enhanced iterative reasoning process combining classification tasks with mathematical calculations at runtime. The ontology as well as the reasoning method were successfully evaluated based on different phenotypes (including SOFA score, socio-economic status, body surface area and WHO BMI classification) and several data sets.

Keywords. Phenotype definition, phenotype classification, phenotype calculation, phenotype ontology, phenotype reasoning

[1] Alexandr Uciteli, IMISE, University of Leipzig, Härtelstraße 16-18, 04107 Leipzig, Germany; E-mail: auciteli@imise.uni-leipzig.de. Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

Despite its long ago introduction in 1909 by Wilhelm Johannsen, the term ‘phenotype’ still has no generally agreed definition [1]. Usually, a phenotype is considered as an observable characteristic or trait of an organism, such as its morphology, function, behaviour, or its biochemical and physiological properties [1–3]. Correct determination of phenotypes plays a key role for diagnosis of diseases, evaluation of risk factors and recruitment of patients for clinical and epidemiological studies [4,5]. One challenge is to translate phenotype algorithms, which “are most commonly represented as non-computable descriptive documents and knowledge artifacts” [6], into machine-readable form. Recent attempts have shown that ontologies are suitable to handle phenotypes and that they can support clinical research and decision making [7–9].

The main goal of the German Medical Informatics Initiative (MII) [10,11] is making clinical data available for research. Most German university hospitals participate in one of the four funded consortia. Smart Medical Information Technology for Healthcare (SMITH) is one of these consortia [12]. Within the ongoing SMITH project, a phenotyping pipeline (PheP) will be established to systematically develop, evaluate and execute validated algorithms and models for classifying and annotating patient data based on routine EHR. These annotations and derivatives will be provided for triggering alerts and actions, data sharing and deep analyses of patient care and outcomes. Phenotype engines and factories are required as an overall infrastructure to specify, set up and execute phenotype algorithms.

In this article, we propose a novel ontology-based method to model and calculate phenotypes. Our approach provides an extended reasoning combining phenotypic data to derive complex phenotypes based on calculations and classifications. The developed tools are designed to work as phenotype engine and factory in SMITH context.

2. Methods

This section outlines the embedding of the PhenoMan in the SMITH infrastructure (**Figure 1**).

The required EHR data will be integrated in a Health Data Storage (HDS) in a standardized manner based on HL7 FHIR [13]. Structured data from different source systems in hospitals as well as unstructured documents are taken into account. Natural Language Processing (NLP) techniques are used to extract and transform relevant data from unstructured EHR documents into structured form. For the specification of the HDS schema (i.e., metadata including single data elements, data element groups, value sets, referenced terminologies, etc.) required to transform and integrate data from various sources, the software ART-DECOR® [14] is used. ART-DECOR® is an open-source tool suite that enables creation and maintenance of HL7 templates, value sets, scenarios and data sets and supports, inter alia, FHIR capabilities.

The PhenoMan imports the data elements from ART-DECOR® and inserts them into the ontology. The phenotype designer uses the Phenotype Editor to develop phenotype algorithms/models based on the source data elements. Each phenotype algorithm is saved as a Phenotype Algorithm Specification Ontology (PASO) by PhenoMan. For the communication with the FHIR Server, the PhenoMan Service is established, which encapsulates the PhenoMan API. The service generates subscriptions (rest-hook) [15] for each PASO and transmits them to the FHIR Server. As soon as FHIR resources (e.g., patient or observation resources) are present that fulfil the criterion of a subscription (e.g., after update or create), the FHIR Server sends the resources to the PhenoMan Service. Additionally, the PhenoMan Service can request further resources (e.g., observations, conditions or medications) required for phenotypes calculation/reasoning. After receiving required resources, the PhenoMan Service calculates phenotypes (using PhenoMan API and PASOs) and writes the results as observation resources back to the FHIR Server. For the specification of the subscription criteria and querying the FHIR Server, FHIR Search [16] is used.

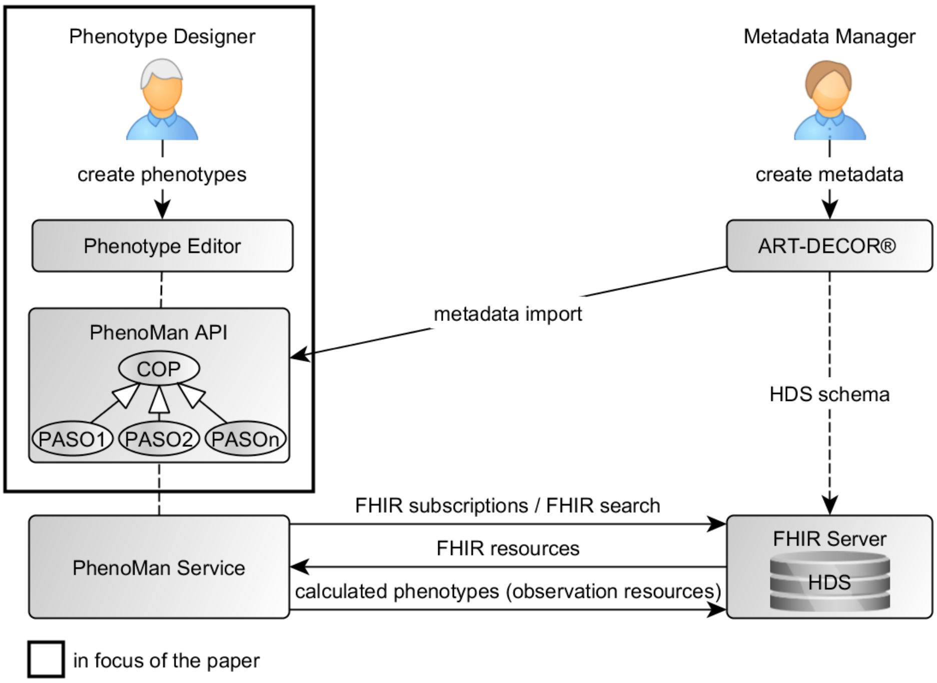


Figure 1. Proposed PheP architecture

This work focusses on the ontology-based modelling and reasoning of phenotypes using PhenoMan. The SMITH infrastructure components as well as the integration of PhenoMan in SMITH will be described in details in further papers.

3. Results

3.1. Core Ontology of Phenotypes (COP)

We developed the Core Ontology of Phenotypes (COP, **Figure 2**) to model, classify and calculate phenotypes based on instance data sets (e.g., of a patient). In this article, we consider a phenotype as an individual (in sense of General Formal Ontology, GFO [17]), for example, the weight of a specific person. Hereinafter, abstract instantiable entities that are instantiated by phenotypes are called phenotype classes. For instance, the abstract property ‘weight’ possess individual weights as instances. We distinguish between single and composite properties (traits), and correspondingly, between single and composite phenotypes. A composite property is defined as a property that has single properties as parts [18]. Based on the definitions of single and composite properties [18], we define *single phenotypes* as single properties (e.g., age, weight, height) and *composite phenotypes* as composite properties (e.g., height and weight, BMI, SOFA score [19]) of an organism[2] or of one of its subsystems. Composite phenotypes are divided into combined and derived phenotypes. A *combined phenotype* is only a combination of corresponding phenotypes (e.g., a combination of height and weight), whereas a *derived phenotype* is an additional property (e.g., BMI) derived from the corresponding phenotypes (height and weight). In the framework of GFO we modelled properties or traits using the class gfo:Property. In the present article, composite phenotype classes are modelled using a Boolean expression based on has_part relation (e.g., weight and height: has_part some height and has_part some weight). Derived phenotype classes additionally define a calculation rule/mathematical formula (e.g., BMI = weight[kg] / height[m]²). Furthermore, combined phenotype classes can associate certain conditions with specific predefined values (scores), which can be used, e.g., in further formulas. For example, if bilirubin value is greater than 12 mg/dl, then the value 4 is used for the calculation of the SOFA score [19].

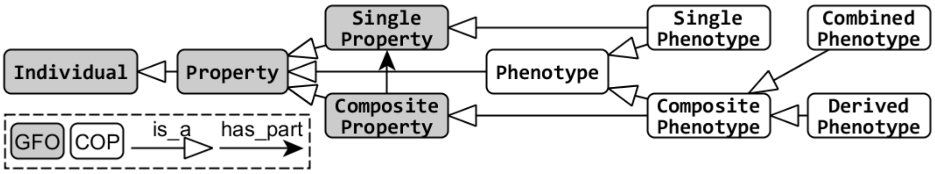


Figure 2. Core Ontology of Phenotypes (COP)

Additionally, we distinguish between restricted and non-restricted phenotype classes, depending on whether their extensions (set of instances) are restricted to a certain range of individual phenotypes by defined conditions or all instances are allowed. For example, the phenotype class ‘age’ is instantiated by the ages of all living beings (non-restricted), whereas the phenotype class ‘young age’ is instantiated by the ages of the young ones, e.g., if the age is below 30 years (restricted).

3.2. Phenotype Algorithm Specification Ontologies (PASO)

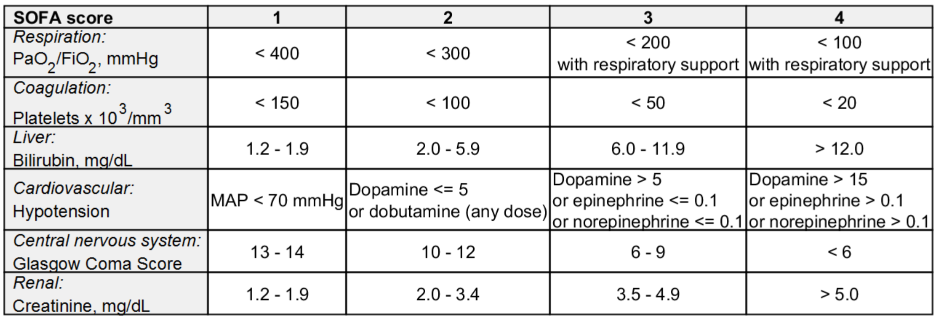
Specific phenotypes (algorithms) are modelled in Phenotype Algorithm Specification Ontologies (PASO)[3] using the COP. PASOs are embedded in the COP in such a way that the classes of the PASO are subclasses of the COP classes. Every PASO subclass of the COP classes cop:Single_Phenotype, cop:Combined_Phenotype or cop:Derived_Phenotype is a phenotype class and is instantiated by phenotypes. The direct subclasses are non-restricted (e.g., Bilirubin, **Figure 4**), while the subclasses of the non-restricted phenotype classes are restricted (e.g., Bilirubin_s_ge_2_0_l_6_0, i.e., bilirubin between 2 and 6 mg/dL).

Phenotype classes possess various common attributes (e.g., labels, descriptions and links to external concepts). Other attributes vary depending on the type of the phenotype

[2] Properties of an organism are considered as all documentable information about it, whereby the modeller is left to decide what is relevant to the current situation.

[3] A PASO is not a usual domain ontology describing a domain by suitable concepts, different relations between them and axioms (like "patient is treated in some hospitals", "patient has some diseases" or "disease was diagnosed by some doctors"). The main purpose of a PASO is to efficiently model concrete phenotypes (algorithms) that should be calculated by the software based on relevant patient characteristics.

class. Non-restricted single phenotype (NSiP) classes, for example, define the datatype, a unit of measure and an optional aggregate function; non-restricted derived phenotype (NDeP) classes – a mathematical formula; restricted single (RSiP) and derived phenotype (RDeP) classes – a restriction; and restricted combined phenotype (RCoP) classes – an optional score value. The logical relations between phenotype classes as well as range restrictions are represented in OWL by anonymous equivalent classes or general class axioms based on property restrictions.



SOFA score	1	2	3	4
Respiration: PaO_2/FiO_2, mmHg	< 400	< 300	< 200 with respiratory support	< 100 with respiratory support
Coagulation: Platelets x $10^3/mm^3$	< 150	< 100	< 50	< 20
Liver: Bilirubin, mg/dL	1.2 - 1.9	2.0 - 5.9	6.0 - 11.9	> 12.0
Cardiovascular: Hypotension	MAP < 70 mmHg	Dopamine <= 5 or dobutamine (any dose)	Dopamine > 5 or epinephrine <= 0.1 or norepinephrine <= 0.1	Dopamine > 15 or epinephrine > 0.1 or norepinephrine > 0.1
Central nervous system: Glasgow Coma Score	13 - 14	10 - 12	6 - 9	< 6
Renal: Creatinine, mg/dL	1.2 - 1.9	2.0 - 3.4	3.5 - 4.9	> 5.0

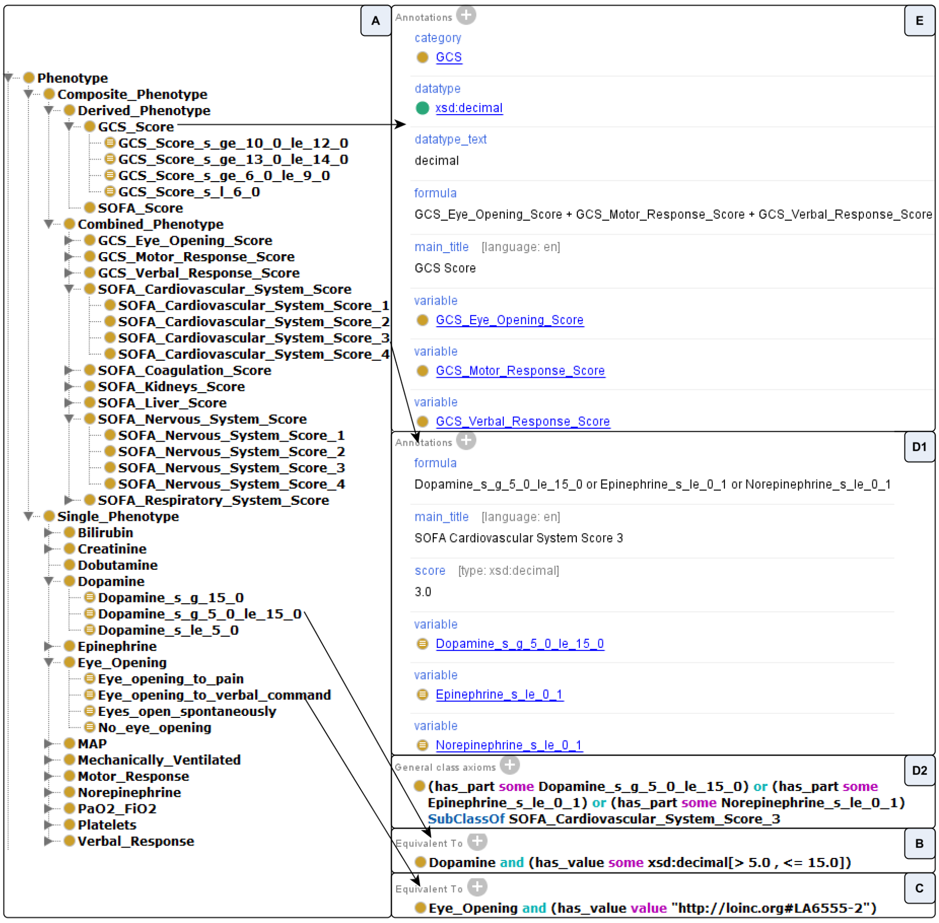
Figure 3. SOFA score [19]

The modelling procedure is illustrated by means of an example for calculating the SOFA (Sequential (or Sepsis-related) Organ Failure Assessment) score [19]. The SOFA score plays an important role in medicine to quantitatively describe the degree of multiple organ dysfunction/failure over time in patients. The total score is calculated as a sum of the 6 single organ scores (respiration, coagulation, liver, cardiovascular, central nervous system and renal). Each single organ score may take values from 0 (normal) to 4 (most abnormal), so that the maximum SOFA score is 24 (**Figure 3**).

First, we model the NSiP classes, e.g., Bilirubin, Dopamine and Eye_Opening representing single patient characteristics relevant for calculating the SOFA score as subclasses of cop:Single_Phenotype (**Figure 4**). Labels, descriptions, related concepts, etc. can be specified as annotations. Next, the RSiP classes (e.g., Bilirubin_s_ge_2_0_l_6_0, Dopamine_s_g_5_0_le_15_0 or Eye_opening_to_verbal_command) for value ranges are defined as subclasses of the NSiP classes. For every RSiP class, the anonymous equivalent class is created that represents the corresponding restriction (**Figure 4**: B, C). The single organ scores can be modelled using combined phenotype classes. For each score a subclass of cop:Combined_Phenotype is defined (e.g., SOFA_Liver_Score, SOFA_Cardiovascular_System_Score or GCS_Eye_Opening_Score). The subclasses of these non-restricted combined phenotype (NCoP) classes represent the single score values (e.g., SOFA_Cardiovascular_System_Score_3). These classes reference the corresponding RSiP range classes using a general class axiom and define the score values (**Figure 4**: D1, D2).

The score for nervous system, the Glasgow Coma Scale (GCS) [20], is calculated as a sum of three single scores “Eye opening”, “Verbal response” and “Motor response”. We model the GCS as a NDeP class. The formula is defined as annotation using the names of NCoP classes (**Figure 4**: E). Now, the RDeP classes for GCS ranges are defined (e.g., GCS_Score_s_ge_10_0_le_12_0). Then, the overall nervous system score is modelled as NCoP class SOFA_Nervous_System_Score with RCoP classes (e.g., SOFA_Nervous_System_Score_3), which reference the GCS range classes and define the score values.

The final step is to define the SOFA score as NDeP class and to specify the formula 'SOFA_Cardiovascular_System_Score + SOFA_Coagulation_Score + SOFA_Kidneys_Score + SOFA_Liver_Score + SOFA_Nervous_System_Score + SOFA_Respiratory_System_Score'.



- Phenotype
  - Composite_Phenotype
    - Derived_Phenotype
      - GCS_Score
        - GCS_Score_s_ge_10_0_le_12_0
        - GCS_Score_s_ge_13_0_le_14_0
        - GCS_Score_s_ge_6_0_le_9_0
        - GCS_Score_s_l_6_0
      - SOFA_Score
    - Combined_Phenotype
      - GCS_Eye_Opening_Score
      - GCS_Motor_Response_Score
      - GCS_Verbal_Response_Score
      - SOFA_Cardiovascular_System_Score
        - SOFA_Cardiovascular_System_Score_1
        - SOFA_Cardiovascular_System_Score_2
        - SOFA_Cardiovascular_System_Score_3
        - SOFA_Cardiovascular_System_Score_4
      - SOFA_Coagulation_Score
      - SOFA_Kidneys_Score
      - SOFA_Liver_Score
      - SOFA_Nervous_System_Score
        - SOFA_Nervous_System_Score_1
        - SOFA_Nervous_System_Score_2
        - SOFA_Nervous_System_Score_3
        - SOFA_Nervous_System_Score_4
      - SOFA_Respiratory_System_Score
  - Single_Phenotype
    - Bilirubin
    - Creatinine
    - Dobutamine
    - Dopamine
      - Dopamine_s_g_15_0
      - Dopamine_s_g_5_0_le_15_0
      - Dopamine_s_le_5_0
    - Epinephrine
    - Eye_Opening
      - Eye_opening_to_pain
      - Eye_opening_to_verbal_command
      - Eyes_open_spontaneously
      - No_eye_opening
    - MAP
    - Mechanically_Ventilated
    - Motor_Response
    - Norepinephrine
    - PaO2_FiO2
    - Platelets
    - Verbal_Response

Annotations

category
GCS

datatype
xsd:decimal

datatype_text
decimal

formula
GCS_Eye_Opening_Score + GCS_Motor_Response_Score + GCS_Verbal_Response_Score

main_title [language: en]
GCS Score

variable
GCS_Eye_Opening_Score

variable
GCS_Motor_Response_Score

variable
GCS_Verbal_Response_Score

Annotations

formula
Dopamine_s_g_5_0_le_15_0 or Epinephrine_s_le_0_1 or Norepinephrine_s_le_0_1

main_title [language: en]
SOFA Cardiovascular System Score 3

score [type: xsd:decimal]
3.0

variable
Dopamine_s_g_5_0_le_15_0

variable
Epinephrine_s_le_0_1

variable
Norepinephrine_s_le_0_1

General class axioms

(has_part some Dopamine_s_g_5_0_le_15_0) or (has_part some Epinephrine_s_le_0_1) or (has_part some Norepinephrine_s_le_0_1) SubClassOf SOFA_Cardiovascular_System_Score_3

Equivalent To

Dopamine and (has_value some xsd:decimal[> 5.0 , <= 15.0])

Equivalent To

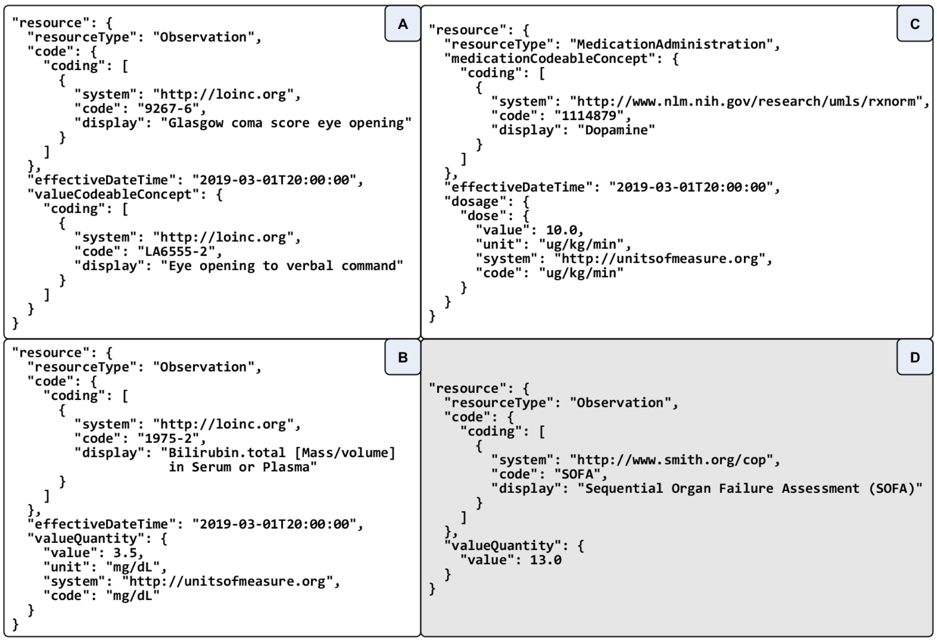
Eye_Opening and (has_value value "http://loinc.org#LA6555-2")

Figure 4. Parts of the SOFA PASO in Protégé

3.3. Phenotype Manager (PhenoMan)

We developed the software Phenotype Manager (PhenoMan), which implements a multistage reasoning approach combining standard reasoners (e.g., Pellet or HermiT) and mathematical calculations. This section briefly outlines the main ideas of our solution based on the example from section 3.2.

First, an instance data set received from the FHIR Server as FHIR resources (**Figure 5**: A-C) is interpreted by PhenoMan and inserted into the ontology. On the one hand, the individual properties (single phenotypes) are inserted as instances of the direct subclasses of cop:Single_Phenotype (Bilirubin, Dopamine, Eye_Opening, etc.) and the values are modelled as property assertions based on the has_value relation (e.g., “has_value 10” for Dopamine). On the other hand, a composite phenotype is defined as instance of the class cop:Composite_Phenotype, which combines all the single phenotype instances using property assertions based on has_part relation. In the first step (classification step), a standard reasoner classifies the single phenotype instances in restricted classes. In our example, the instance of Eye_Opening is classified in the class Eye_opening_to_verbal_command, the instance of Bilirubin – in the class Bilirubin_s_ge_2_0_l_6_0 (i.e., the Bilirubin value is >= 2.0 and < 6.0 mg/dL), the instance of Dopamine – in the class Dopamine_s_g_5_0_le_15_0, etc.



A

```
"resource": {
  "resourceType": "Observation",
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "9267-6",
        "display": "Glasgow coma score eye opening"
      }
    ]
  },
  "effectiveDateTime": "2019-03-01T20:00:00",
  "valueCodeableConcept": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "LA6555-2",
        "display": "Eye opening to verbal command"
      }
    ]
  }
}
```

C

```
"resource": {
  "resourceType": "MedicationAdministration",
  "medicationCodeableConcept": {
    "coding": [
      {
        "system": "http://www.nlm.nih.gov/research/umls/rxnorm",
        "code": "1114879",
        "display": "Dopamine"
      }
    ]
  },
  "effectiveDateTime": "2019-03-01T20:00:00",
  "dosage": {
    "dose": {
      "value": 10.0,
      "unit": "ug/kg/min",
      "system": "http://unitsofmeasure.org",
      "code": "ug/kg/min"
    }
  }
}
```

B

```
"resource": {
  "resourceType": "Observation",
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "1975-2",
        "display": "Bilirubin.total [Mass/volume]
                    in Serum or Plasma"
      }
    ]
  },
  "effectiveDateTime": "2019-03-01T20:00:00",
  "valueQuantity": {
    "value": 3.5,
    "unit": "mg/dL",
    "system": "http://unitsofmeasure.org",
    "code": "mg/dL"
  }
}
```

D

```
"resource": {
  "resourceType": "Observation",
  "code": {
    "coding": [
      {
        "system": "http://www.smith.org/cop",
        "code": "SOFA",
        "display": "Sequential Organ Failure Assessment (SOFA)"
      }
    ]
  },
  "valueQuantity": {
    "value": 13.0
  }
}
```

Figure 5. FHIR-JSON example (A-C: input resources; D: output resource)

A: The value of the “Glasgow coma score eye opening” (LONC: 9267-6) observation is “Eye opening to verbal command” (LOINC: LA6555-2).

B: The value of the Bilirubin (LONC: 1975-2) observation is 3.5 mg/dL.

C: The dose of the medication administration of Dopamine (RxNorm: 1114879) is 10 µg/kg/min.

D: The SOFA score calculated by PhenoMan is 13.

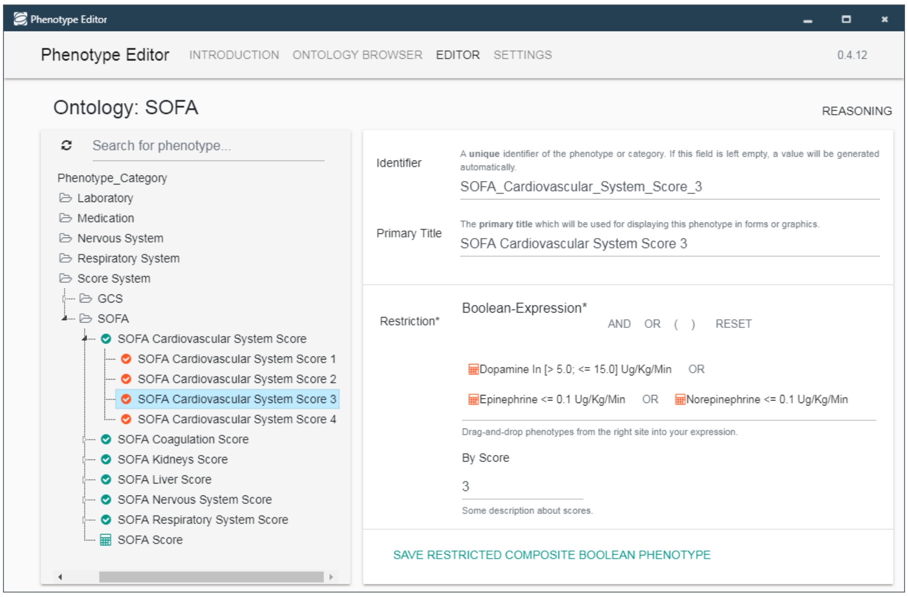
Next, the composite phenotype instance is classified in the suitable score value classes. For instance the cardiovascular system score has the score value 3, because the composite phenotype instance is classified in the class SOFA_Cardiovascular_System_Score_3 (**Figure 4**: D1, D2). In the next step (calculation step), the formula of the derived phenotype class GCS_Score can be calculated by PhenoMan. It inserts the determined score values for “Eye opening”, “Verbal response” and “Motor response” in the formula and calculated the sum. After the calculation the classification step must be performed again. The GCS_Score instance is classified in the class GCS_Score_s_ge_10_0_le_12_0, so that the score value of the nervous system score can be determined. In the final calculation step, the overall SOFA score value is calculated based on the six single organ scores.

In the case of complex phenotypes (e.g., SOFA) the classification and calculation steps can be executed several times. That is the case if a NDeP class has subclasses, i.e., RDeP classes, which are in turn used in combined phenotypes. Both steps are repeated until all formulas are calculated and all phenotypes are classified. Then, all derived and calculated phenotypes are returned by PhenoMan as FHIR resources (**Figure 5**: D).

The PhenoMan supports 4 primitive datatypes xsd:decimal, xsd:string, xsd:boolean and xsd:date. All other complex datatypes (e.g., FHIR code or quantity) are mapped to the primitive datatypes (e.g., code to xsd:string with additional attributes and quantity to xsd:decimal with additional unit attribute). Furthermore, the PhenoMan provides, inter alia, aggregate functions, Boolean, date and measurement unit arithmetic, integration of external terminologies as well as reading and writing FHIR resources. Nevertheless, it is not our aim to completely model the EHR. Instead, our approach can support the modelling and calculation of selected phenotypes in a user-friendly standardized manner.

3.4. Phenotype Editor

The Phenotype Editor is an interactive user interface for managing and developing PASOs. In **Figure 6** you can see how the phenotype SOFA_Cardiovascular_System_Score_3 is defined with the Phenotype Editor forms. The phenotype is a restricted combined phenotype and thus, requires a Boolean expression, which was built by drag-and-dropping the phenotypes from the left site into the expression form field. The form data is transferred to the backend service via JSON and the service uses the PhenoMan API to insert the phenotype metadata into a PASO.



Ontology: SOFA

Identifier: A **unique** identifier of the phenotype or category. If this field is left empty, a value will be generated automatically.
SOFA_Cardiovascular_System_Score_3

Primary Title: The **primary title** which will be used for displaying this phenotype in forms or graphics.
SOFA Cardiovascular System Score 3

Restriction*: Boolean-Expression*
Dopamine In [> 5.0; <= 15.0] Ug/Kg/Min OR
Epinephrine <= 0.1 Ug/Kg/Min OR Norepinephrine <= 0.1 Ug/Kg/Min
Drag-and-drop phenotypes from the right site into your expression.

By Score
3
Some description about scores.

Figure 6. Screenshot of the Phenotype Editor. We left out some of the metadata fields for better visibility.

3.5. Implementation

The PhenoMan is implemented in Java using OWL API [21] and two reasoners, HermiT [22] and Openllet [23]. For calculations we utilize the Java Expression Evaluator (EvalEx) [24], but the integration of other libraries (e.g., for executing R scripts) or rule systems (e.g., SWIRL or Drools) is also possible. The EvalEx enables evaluating mathematical and Boolean (inter alia, Boolean operators and IF-THEN-ELSE structures) expressions and supports defining custom functions and operators.

The Phenotype Editor[4] is a desktop app, designed with JavaScript and is shipped as cross platform Electron [25] app with an integrated lightweight web browser (Chromium). We decided to outsource the logic (i.e., creation/update of a phenotype and reasoning) into a backend service[5], which provides information and management functionalities of a PASO via REST interface. The backend is a DropWizard [26] application, which serves as a mediator to the PhenoMan API. The advantage of splitting the phenotype managing application into frontend and backend is, that users are able to work on one ontology collaboratively and all created ontologies are centrally stored. The ontology service could also be executed on the local machine, so that the user could use it to create his own ontologies. Additional features like access control or audit logging are currently not available, but we plan to add them in future releases.

4. Related Work

We developed a novel approach to support ontological modelling and reasoning of phenotypes. In contrast to [7,8], our solution serves to determine and to classify phenotypes based on instance data (e.g., EHR). Moreover, the proposed reasoning process includes calculation of mathematical formulas at runtime.

Very similar to our approach, Fernández-Breis et al. [27] propose to take advantage of the best features of EHR standards and ontologies. The authors developed methods allowing a direct use of EHR data for the identification of patient cohorts leveraging current EHR standards and semantic web technologies. In [27], openEHR [28] archetypes were used as EHR standard. An ontological infrastructure was designed including different ontologies for representing domain entities (colorectal-domain), the rules for determining the risk level and the data. The mappings between the phenotyping archetype and the colorectal-domain ontology were defined and are automatically executed on the archetyped data instances to generate the OWL dataset. The data is then transformed into OWL, where the classification is performed. We use HL7 FHIR as a standard for exchanging healthcare information in the SMITH infrastructure. But the main difference to the approach of Fernández-Breis et al. lies in our three-level ontological architecture. The COP is founded by GFO and provides a framework for developing PASOs. In this way, each particular phenotype algorithm specified as a PASO has the same standardized structure and can be executed by PhenoMan in the same manner. A further advantage of our solution is that the PhenoMan supports classification as well as calculation tasks and works directly with FHIR format, so that no further transformations are required. The mapping between EHR data and ontology is performed by PhenoMan automatically using terminology associations, which are defined for each data element in ART-DECOR® (and imported into ontology) as well as in FHIR resources (e.g., Observation).

The main objective of SHARPn [29] is to develop methods and modular open-source resources for enabling secondary use of EHR data for high-throughput phenotyping. The

[4] Source code and releases of the Phenotype Editor are available on GitHub under the GPL-3.0 license: https://github.com/ChristophB/phenotype_editor

[5] Source code and releases of the Ontology Service (backend) are available on GitHub under the GPL-3.0 license: https://github.com/ChristophB/ontology_service

phenotype algorithms are specified based on Quality Data Model (QDM) [30] and represented in the HL7 Health Quality Measures Format (HQMF or eMeasure) [31]. According to the authors, there are two main challenges. Firstly, data elements in an EHR may not be represented in a format consistent with the QDM. Secondly, an EHR typically does not natively have the capability to automatically consume and execute eMeasure logic. To address these challenges, a translator tool was developed that converts QDM-defined phenotyping algorithm criteria into executable Drools rules scripts.

The Phenotype Execution and Modeling Architecture (PhEMA) [32] is an open-source infrastructure for standards-based authoring, sharing, and execution of phenotyping algorithms. Similarly to SHARPn, PhEMA uses QDM and HQMF to model phenotype definitions. Phenotyping algorithms are represented using the PhEMA Authoring Tool (PhAT), are exported from the PhAT into executable KNIME [33] workflows and are executed against data warehouses or data repositories.

In contrast to the rule- or workflow-based description of phenotyping algorithms, we use an ontology-based one. Our approach is rather generic and enables a standardized and structured modelling as well as the reuse of phenotyping algorithms and their parts (e.g., concepts and restrictions). Furthermore, the PhenoMan is compatible with the native representation of EHR data (HL7 FHIR) in the SMITH infrastructure and does not need an additional import of the data into a data warehouse.

In [34] a FHIR-compatible model was designed to support capture of cancer clinical data. Our approach allows the modelling of different phenotypes based on a core ontology (COP) and is independent of the EHR representation standards. The interpretation of FHIR data and the mapping to specified phenotypes using terminology associations are provided by PhenoMan.

A method to enable automated transformation of clinical data into OWL ontologies is presented in [35]. The developed system generates OWL representations of openEHR archetypes and automatically transforms openEHR data to OWL individuals. In our approach, the phenotypes are directly modelled in the ontology and are automatically mapped to the EHR data. Moreover, our solution supports classification as well as calculation of phenotypes.

As described in section 3.1, phenotypes can possess links to concepts of external ontologies. For instance, they may be annotated with concepts of anatomic structures (e.g., Foundational Model of Anatomy [36]), or situations, respective processes, where phenotypes are observed (e.g., electrocardiographic monitoring). The linkage is similar to the Entity-Quality method [37] (entity: anatomic structure or process, quality: phenotype) and may improve comparison of COP across multiple domains.

Hoehndorf et al. [8] proposed the PhenomeNET for incorporation of phenotype ontologies from different species. PhenomeNET can predict orthologous genes with common pathways and common related diseases. Apart from the different interpretation of the term ‘phenotype’, the main focus of our attempt is to deduce complex phenotypes from a set of basic phenotypes of an individual.

The Human Phenotype Ontology (HPO) [7] associates phenotypic abnormalities with underlying diseases and participating genes, whereas COP can contain all sorts of properties of an organism (including non-abnormalities). Currently, COP does not offer weights for phenotype-disease relations, like HPO does to sort diseases for a phenotype set by relevance. We will investigate ways to add this functionality to COP in future.

5. Conclusion and Future Work

We developed a novel ontology-based method to model phenotypes of living beings with the aim of automated phenotype reasoning based on instance data (e.g., patient data). Our solution includes an enhanced reasoning process, which is iterative and combines classification tasks with mathematical calculations at runtime. This new approach can be used in clinical context, e.g., for supporting the diagnostic process, evaluating risk factors or recruiting appropriate participants for clinical or epidemiological studies. About 20 phenotype algorithms have already been modelled and the ontology as well as the reasoning method were successfully evaluated based on several data sets. Some algorithms (such as socio-economic status[6], SES [38]) were evaluated in comparison with the corresponding SPSS derivatives based on the research database of the LIFE study [39].

An integration of more complex algorithms into the reasoning process is possible and has to be investigated in respect of accessing external libraries (e.g., R scripts). The current formalism will be extended in the future to include the further desiderata expounded by Mo et al. [6]. PhenoMan and Phenotype Editor will function as phenotype engine and factory in SMITH context.

6. Conflict of Interest

The authors state that they have no conflict of interests.

7. Acknowledgment

This work was supported by the German Federal Ministry of Education and Research (SMITH: 01ZZ1803A; Leipzig Health Atlas: 031L0026).

References

[1] M. Mahner, and M. Kary, What exactly are genomes, genotypes and phenotypes? And what about phenomes?, *J. Theor. Biol.* **186** (1997) 55–63.

[2] R. Hoehndorf, A. Oellrich, and D. Rebholz-Schuhmann, Interoperability between phenotype and anatomy ontologies, *Bioinformatics*. **26** (2010) 3112–3118. doi:10.1093/bioinformatics/btq578.

[3] A. Uciteli, S. Groß, S. Kireyev, and H. Herre, An ontologically founded architecture for information systems in clinical and epidemiological research, *J. Biomed. Semant.* **2** (2011) S1.

[4] A.R. Deans, S.E. Lewis, E. Huala, et al., Finding Our Way through Phenotypes, *PLOS Biol.* **13** (2015) e1002033.

[5] P.N. Robinson, Deep phenotyping for precision medicine, *Hum. Mutat.* **33** (2012) 777–780.

[6] H. Mo, W.K. Thompson, L.V. Rasmussen, et al., Desiderata for computable representations of electronic health records-driven phenotype algorithms, *J. Am. Med. Inform. Assoc. JAMIA*. **22** (2015) 1220–1230.

[7] S. Köhler, N.A. Vasilevsky, M. Engelstad, et al., The Human Phenotype Ontology in 2017, *Nucleic Acids Res.* **45** (2017) D865–D876.

[8] R. Hoehndorf, P.N. Schofield, and G.V. Gkoutos, PhenomeNET: a whole-phenome approach to disease gene discovery, *Nucleic Acids Res.* **39** (2011) e119.

[6] We consider SES in the broadest sense as a composite property of a person, i.e., as a kind of derivative that can be modelled and calculated in the same way as a phenotype.

[9] F. Loebe, F. Stumpf, R. Hoehndorf, and H. Herre, Towards improving phenotype representation in OWL, *J. Biomed. Semant.* **3** (2012) S5. doi:10.1186/2041-1480-3-S2-S5.

[10] S. Gehring, and R. Eulenfeld, German Medical Informatics Initiative: Unlocking Data for Research and Health Care, *Methods Inf. Med.* **57** (2018) e46–e49. doi:10.3414/ME18-13-0001.

[11] S.C. Semler, F. Wissing, and R. Heyder, German Medical Informatics Initiative, *Methods Inf. Med.* **57** (2018) e50–e56. doi:10.3414/ME18-03-0003.

[12] A. Winter, S. Stäubert, D. Ammon, et al., Smart Medical Information Technology for Healthcare (SMITH), *Methods Inf. Med.* **57** (2018) e92–e105. doi:10.3414/ME18-02-0004.

[13] HL7 FHIR, (n.d.). https://www.hl7.org/fhir/.

[14] ART-DECOR®, (n.d.). https://www.art-decor.org/.

[15] FHIR Subscription, (n.d.). https://www.hl7.org/fhir/subscription.html.

[16] FHIR Search, (n.d.). https://www.hl7.org/fhir/search.html.

[17] H. Herre, General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling, in: R. Poli, M. Healy, and A. Kameas (Eds.), Theory Appl. Ontol. Comput. Appl., Springer, Netherlands, 2010: pp. 297–345.

[18] A. Uciteli, J. Neumann, K. Tahar, et al., Ontology-based specification, identification and analysis of perioperative risks, *J. Biomed. Semant.* **8** (2017) 36.

[19] J.-L. Vincent, R. Moreno, J. Takala, et al., The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure, *Intensive Care Med.* **22** (1996) 707–710.

[20] G. Teasdale, and B. Jennett, ASSESSMENT OF COMA AND IMPAIRED CONSCIOUSNESS: A Practical Scale, *The Lancet*. **304** (1974) 81–84.

[21] OWL API, (n.d.). http://owlcs.github.io/owlapi/.

[22] HermiT Reasoner, (n.d.). http://www.hermit-reasoner.com/.

[23] Openllet Reasoner, 2018. https://github.com/Galigator/openllet.

[24] U. Klimaschewski, EvalEx - Java Expression Evaluator, 2019. https://github.com/uklimaschewski/EvalEx.

[25] Electron, (n.d.). https://electronjs.org/.

[26] Dropwizard, (n.d.). https://www.dropwizard.io.

[27] J.T. Fernández-Breis, J.A. Maldonado, M. Marcos, M. del C. Legaz-García, D. Moner, J. Torres-Sospedra, A. Esteban-Gil, B. Martínez-Salvador, and M. Robles, Leveraging electronic healthcare record standards and semantic web technologies for the identification of patient cohorts, *J. Am. Med. Inform. Assoc.* **20** (2013) e288–e296.

[28] openEHR, (n.d.). https://www.openehr.org/.

[29] J. Pathak, K.R. Bailey, C.E. Beebe, et al., Normalization and standardization of electronic health records for high-throughput phenotyping: the SHARPn consortium, *J. Am. Med. Inform. Assoc. JAMIA*. **20** (2013) e341–348.

[30] QDM - Quality Data Model, (n.d.). https://ecqi.healthit.gov/qdm-quality-data-model.

[31] HQMF - Health Quality Measure Format, (n.d.). https://ecqi.healthit.gov/hqmf-health-quality-measure-format.

[32] J.A. Pacheco, L.V. Rasmussen, R.C. Kiefer, T.R. Campion, P. Speltz, R.J. Carroll, S.C. Stallings, H. Mo, M. Ahuja, G. Jiang, E.R. LaRose, P.L. Peissig, N. Shang, B. Benoit, V.S. Gainer, K. Borthwick, K.L. Jackson, A. Sharma, A.Y. Wu, A.N. Kho, D.M. Roden, J. Pathak, J.C. Denny, and W.K. Thompson, A case study evaluating the portability of an executable computable phenotype algorithm across multiple institutions and electronic health record environments, *J. Am. Med. Inform. Assoc. JAMIA*. **25** (2018) 1540–1546.

[33] KNIME, (n.d.). https://www.knime.com/.

[34] H. Hochheiser, M. Castine, D. Harris, G. Savova, and R.S. Jacobson, An information model for computable cancer phenotypes, *BMC Med. Inform. Decis. Mak.* **16** (2016) 121.

[35] B. Haarbrandt, T. Jack, and M. Marschollek, Automated Transformation of openEHR Data Instances to OWL, *Stud. Health Technol. Inform.* **223** (2016) 63–70.

[36] C. Rosse, and J.L.V. Mejino, A reference ontology for biomedical informatics: the Foundational Model of Anatomy, *J. Biomed. Inform.* **36** (2003) 478–500.

[37] N.L. Washington, M.A. Haendel, C.J. Mungall, M. Ashburner, M. Westerfield, and S.E. Lewis, Linking human diseases to animal models using ontology-based phenotype annotation, *PLoS Biol.* **7** (2009) e1000247.

[38] T. Lampert, S. Müters, H. Stolzenberg, L.E. Kroll, and KiGGS Study Group, Measurement of socioeconomic status in the KiGGS study: first follow-up (KiGGS Wave 1), *Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz*. **57** (2014) 762–770.

[39] LIFE Health Study - University of Leipzig, (n.d.). http://life.uni-leipzig.de/en/life_health_study.html.