Virtual Coaching Activities for Rehabilitation in Elderly

Call: H2020-SC1-2016-2017

Grant Agreement Number: 769807

Deliverable

D4.3: Baseline ontology, ontology meta models and frameworks SW component, final release

Deliverable type: Other

WP number and title: WP4: Knowledge Integration

Dissemination level: Confidential

Due date: M34

Lead beneficiary: FZI

Lead author(s): Patrick Philipp (FZI)

Co-Author(s): Lars Böcking (FZI)

Reviewers: Massimo Caprino (CCP), Kai Gand (TUD), Hannes Schlieter (TUD)

Document history

Version	Date	Author/Editor	Description
0.1	10.06.2020	Lars Böcking	Draft adjustments based on D4.2
0.2	22.06.2020	Lars Böcking	TechLab alignments included
0.3	23.06.2020	Patrick Philipp	FZI-Internal review of overall story and alignment
0.4	24.06.2020	Lars Böcking Patrick Philipp	First final version for project- internal review
0.5	27.06.2020	Massimo Caprino Kai Gand	Reviewed Version
1.0	29.06.2020	Lars Böcking Patrick Philipp	Final Version

Table of Contents

List of Figures

LIST OF TABLES

LIST OF ACRONYMS

Acronym	Meaning
FOAF	Friend-of-a-Friend
SSN	Semantic Sensor Network Ontology
M3	Machine-to-Machine Measurement
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
N3	Notation3
OWL	Web Ontology Language
SHACL	Shapes Constraint Language
SWT	Semantic Web technologies
MeSH	Medical Subject Headings
SOSA	Sensor, Observation, Sample, and Actuator
ARAT	Action Research Arm Test
IoT	Internet of Things
ML	Machine Learning
RL	Reinforcement Learning

EXECUTIVE SUMMARY

The deliverable presents the final version of the vCare ontology, comprising models and meta-models for representing knowledge in vCare, and is an updated version of **D4.2**.

The vCare ontology is modelled with the Resource Description Framework (RDF), thereby building on Semantic Web technologies (SWT) as proposed in Deliverable **D4.1**. By using RDF, the vCare ontology is directly linked to other relevant and important ontologies and structured knowledge bases (i.e. knowledge graphs), such as (i) RadLex, Gene Ontology, Logical Observation Identifiers Names and Codes, Human Disease Ontology, universAAL Health Measurement, Stroke Ontology or Medical Subject Headings for the medical domain, (ii) Semantic Sensor Network Ontology or Machine-to-Machine Measurement for the Internet-of-Things (IoT) domain, and (iii) Friend-of-a-Friend, vCard or Product ontology for generally applicable concepts and relations. The linking procedure enables to reuse and extend available domain knowledge, which is especially important for the medical domain, as we are able to directly reuse rich, structured models of medical diseases, where causes and possible treatment are documented.

In addition to RDF, the vCare ontology builds on using SHACL, the Shapes Constraint Language, to foster a sustainable vCare ontology schema with correct instances. SHACL enables to define important constraints with respect to required information and possible values or ranges. To this end, the vCare ontology also builds on Notation3 (N3), in order to enable inference of novel information based on expert rules (e.g. evidence indicators), as introduced in **D1.3**.

The deliverable dwells on the evolution from the medical ontology outline from Deliverable **D1.3** and the initial version of the vCare ontology schema as presented in **D4.2**. We find that all classes of the medical ontology outline can be mapped to the vCare ontology, where we use the terminology of Profiles analogously. The feedback class will be explicitly modelled during the Tech Lab phase for which close collaboration with medical experts will refine the possible feedback mechanisms for patients.

This deliverable presents the final version of the vCare ontology, including all adjustments and extensions made to the initial version of the ontology. The deliverable also includes a final version of the vCare ontology with specialized vCare profiles for all use cases and diseases. This final version is based on extensive collaboration with technical- and medical experts in increasingly complex and challenging use case implementation, such that initial profile schemas were finalized (and thus comprise all necessary information) and respective profile instances (i.e. for all vCare use cases and diseases) are available. The final version of the vCare ontology is available under:

https://wisecloud.wiwi.tu-dresden.de/s/UVYqbbsIXpUjaYc

For instance, the implementation of the rule-based Machine Learning agent (based on rules and evidence indicators presented in **D1.3**) where a single activity (e.g. the choice of an adequate treatment activity, such as a serious game or memory training) needs to be personalized, all vCare profiles, and their vCare ontology schemas and instances are updated. The successful implementation enables us to easily model finalized vCare profiles. Further

information about the implementation of the smart agent and the pattern recognition can be found in the upcoming deliverable **D4.4**.

In order to enable adaptive decision-making for the Virtual Coach, such as clinical pathway adaptation based on patient-specific preferences, the vCare ontology also models decision-theoretic concepts and relationships. More specifically, next to already mentioned vCare profiles, the state profile models the required information for choosing and parameterizing sustainable activities for individual patients. The required information consist of vCare profiles, such as the patient profile with personal- and clinical state or the device- and environment profiles for further situational information about the patient, but also historical information about the successful (or unsuccessful) recommendation of activities in past days or weeks as well as expert guidelines in the form of expert rules. Integration during the tech lab preparation and test conduction described in **D1.5** and documented in **D1.6** has led to a series of adaptations and extensions of the existing ontology. The changes made are discussed in **chapter 3.3** and can be found in the rdf schema files for the profiles and the sample instances.

1. Introduction

1.1 Document Scope

The document relates to the finalisation of Task T4.2 - Implementation of a baseline ontology, ontology meta models and frameworks for the representation of diseases, impairments and personal treatment plans. The result of the final version of the baseline ontology lays ground to the upcoming living lab phase.

The first release of the ontology in **D4.2** defined how vCare profiles are structured, how they relate to each other, and what information and knowledge is needed. In addition, the document also showed how vCare profiles need to be connected to a decision-theoretic representation of information to enable adaptive decision-making.

The modeling of the information in the form of the ontology defined in **D4.3** provided a solid basis for the subsequent TechLab phase. During this phase, minor adjustments and changes were made to the ontology. These are motivated from two points of view. On the one hand, the integration work with the technical partners made it possible to identify additional information that can be used as a basis for decisions on the later adaptation of the clinical pathway. On the other hand, an intensive exchange of information with physicians has been possible. On the other hand, an intensive exchange with the medical partners of the vCare project has taken place, on the basis of which additional specific information is to be stored. On the one hand, this is about setting limits in the context of adaptation, which is defined by expert rules. On the other hand, it is about providing relevant information for the treating physicians of a patient, who can make more precise decisions by providing important information about the behavior and development of a patient.

The vCare ontology, in general, is used for providing a controlled vocabulary for all relevant concepts and their relations, leading to a sustainable data model which can easily be linked to available and established structured vocabularies (e.g. medical ontologies, such as RadLex[1], universAAL HealthMeasurement[2] or Human Disease Ontology[3]).

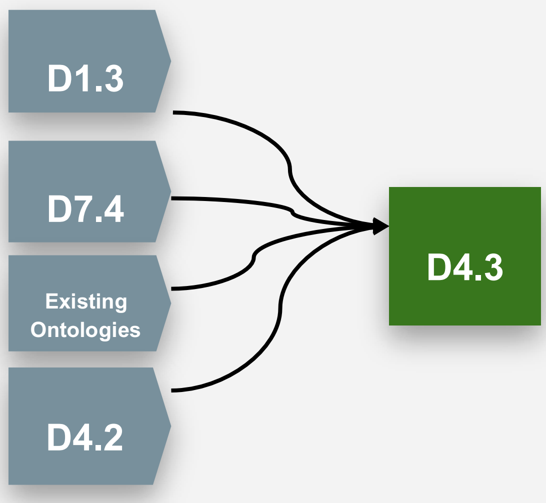

Figure 1: Inputs for the deliverable D4.3.

1.2 DOCUMENT STRUCTURE

The document begins by defining the goal of modelling the vCare ontology by categorizing terminology such as baseline ontology, meta models and frameworks to then integrate the vCare ontology into the broader context of a Virtual Coach (in Section 3). The document then re-captures and summarizes the central modelling approach of the vCare ontology, as dwelled upon in **D4.1**, in order to discuss central design decisions for modelling vCare profiles. Before presenting the final version of the ontology schema, we summarize the existing draft of the vCare ontology as presented in **D4.1**, which was modelled with a collaborative ontology editor in cooperation with medical- and technical experts. In the following chapter we will discuss the adaptation of the ontology within the TechLab phase, which in turn is based on the results of **D4.2**.

The final version of the vCare ontology schema is presented for all vCare profiles, where we also dwell on their mutual relationships. The deliverable is therefore based on contents of **D1.3** and **D7.4**, where, on the one hand, narratives of vCare use cases are linked to outlines of ontologies, and on the other hand, target function- and non-functional requirements of vCare are assigned to different development stages such as the Tech Lab phase, and details of required technical devices are elaborated on. The terminology of *Profiles* for central classes of

[1] https://bioportal.bioontology.org/ontologies/RADLEX
[2] http://ontology.universaal.org/HealthMeasurement.owl
[3] https://bioportal.bioontology.org/ontologies/DOID

the vCare ontology schema is adopted, emphasizing the integration of knowledge from different classes.

We highlight where available knowledge from other ontologies is already integrated on a schema level, which will be further elaborated on for profile instances in Section 4.

Section 4 exemplary instantiates the vCare ontology for different disease rehabilitation based on respective narratives and knowledge representation methods presented in **D1.3**. We further show how integrated external ontologies can be leveraged with respect to knowledge reuse and give an example for vCare state profiles for decision-making.

Section 5 concludes deliverable D4.3 by summarizing the contents of the final version of the vCare ontology and explaining its adequacy for the Living Lab phase.

2. Final vCare Ontology

The section presents the final version of the vCare ontology schema, where all vCare profiles are semantically modelled and mutually related, such that an explicit structure for data and information is available.

Before revisiting our modelling approach, we first need to clarify the used terminology. The *baseline ontology* refers to the general class structure of the vCare ontology, which can be roughly summarized by modelled classes (and their relationships) for vCare profiles. Based on the latter, we need to further specialize the schema (e.g. with respect to different user profiles) and need to model first vCare ontology instances (i.e. instantiations of the vCare ontology schema; e.g. for available types of devices or available types of activities).

Meta-models in the vCare ontology correspond to classes which are either sufficiently abstract that we cannot directly create instances (and need further specialization; e.g. User Profile which specializes to patients or physicians, or Activities which specialize to diagnosis-, treatment- and supporting activities) or classes which aggregate information for analysis and recommendation tasks (e.g. the State Profile, which gathers time-dependent information for patients and their environment). The latter are essential to integrate medical- and technical vocabularies with decision-making vocabularies, but do not add domain specific knowledge to the vCare ontology.

Finally, *software framework* corresponds to the used vCare technology stack, which will be summarized subsequently (Section 3.1).

2.1 Modelling Approach

The schema of the vCare ontology describes a conceptualized view on the computer-assisted patient rehabilitation domain, where the required knowledge in terms of the defined vCare profiles needs to be specified. As baseline for designing and implementing the vCare schema (as well as the instances), we use the Resource Description Framework (RDF)[4] as part of the

[4] https://www.w3.org/TR/rdf11-concepts/

Semantic Web Stack (SWT)[5]. **D4.1** provides an extensive comparison of the SWT and RDF to other knowledge modelling approaches, such as Labelled Property Graphs with Cypher, and concludes that RDF is an eligible choice for the vCare ontology schema. SWT and RDF enable to model complex medical- and technical knowledge as available in vCare, and to connect it with available knowledge in the form of available ontologies and knowledge graphs, such as RadLex[6], Human Disease Ontology (DOID)[7] or Semantic Sensor Ontology[8]. The latter are exemplary ontologies, which model structured information about medical- and technical domains and further ontologies are elaborated on in Section 4. Human Disease Ontology, for instance, structures diseases into hierarchies (e.g. cerebrovascular disease[9], which is a specialization of cardiovascular system disease and vascular disease, links to other ontologies (see *database_cross_reference* for the DOID reference of cerebrovascular disease; e.g. Medical Subject Headings[10] or National Cancer Institute (NCI)[11]) and is being used by other ontologies (e.g. Cardiovascular Disease Ontology[12] or Ontology of Drug Averse Events[13]) which might provide further knowledge.

We additional use the Shapes Constraint Language (SHACL)[14] to ensure integrity of modelled and generated instances of the vCare ontology. SHACL enables to define, in close collaboration with domain experts (i.e. either medical- or technical personnel), which values are permitted for certain types of information (e.g. which levels of difficulty are permitted for a serious game). **D4.1** includes a thorough discussion on the suitability of SHACL, where alternatives such as OWL are compared. SHACL will serve as umbrella framework to model all vCare profiles, which are individual RDF classes with different mandatory and optional relationships to other classes. SHACL governs which relationships are mandatory and is used (when sensible) to constrain the possible values for relationships (e.g. the possible classes or the possible values such as Integers or Strings). By using appropriate SHACL reasoners such as pySHACL[15] of RDFLib[16], one can finally ensure that the modelled vCare ontology instances (e.g. patient profiles for individual patients) are aligned with the vCare ontology schema.

Notation3 (N3)[17] enables us to directly apply modelled expert rules, as introduced in **D1.3**, to derive further knowledge from modelled vCare profiles, such as inferring values for evidence indicators from the patient profile. N3 describes a RDF-based approach to model (among

[5] https://www.w3.org/standards/semanticweb/
[6] https://bioportal.bioontology.org/ontologies/RADLEX
[7] https://bioportal.bioontology.org/ontologies/DOID
[8] https://www.w3.org/TR/vocab-ssn/
[9] http://www.ontobee.org/ontology/DOID?iri=http://purl.obolibrary.org/obo/DOID_6713
[10] https://www.ncbi.nlm.nih.gov/mesh/?term=D002561
[11] https://ncit.nci.nih.gov/ncitbrowser/ConceptReport.jsp?dictionary=NCI_Thesaurus&code=C3390
[12] http://www.ontobee.org/ontology/CVDO
[13] http://www.ontobee.org/ontology/ODAE
[14] https://www.w3.org/TR/shacl/
[15] https://github.com/RDFLib/pySHACL
[16] https://github.com/RDFLib
[17] https://www.w3.org/TeamSubmission/n3/

others) implication rules, where knowledge modelled in RDF can be used as a condition in order to infer novel facts. Such implication rules can be easily modelled as "if-else" statements, which can be applied to medical formulations of rules for evidence indicators or exceptional procedures.

In practical terms, we use the Turtle serialization[18] to model the vCare ontology, as it is easily understandable based on its syntax. Turtle is a specific serialization or format for RDF (i.e. an approach to model the RDF graph structure), which can be translated to any other RDF serialization and is widely used.

2.2 Design Choices for Schema

A conceptual overview of the final version of the vCare ontology schema is presented in **Figure 2**, which is identical to the schema of **D4.2**, as the Tech Lab activities only unveiled the need to adapt

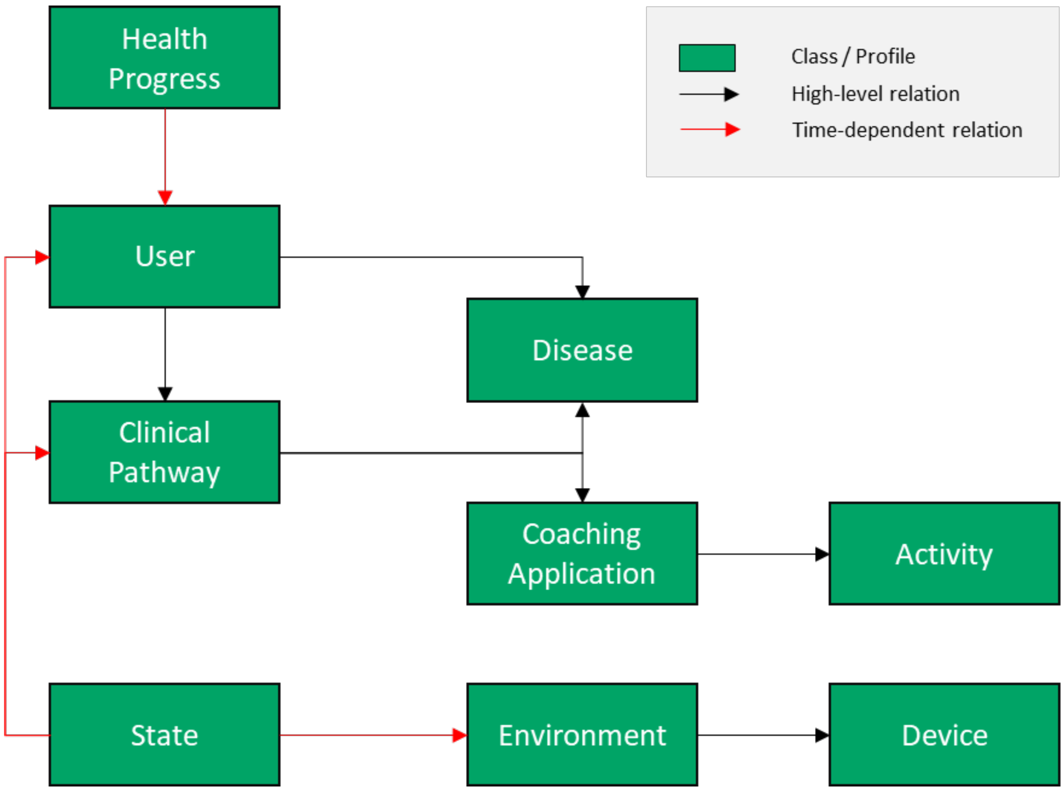

Figure 2: vCare Ontology Schema Overview

The overview illustrates the core vCare classes and their high-level relationships with the aim to integrate all available vCare-related information. The green rectangles represent classes and profiles, which are connected via their relationships. Black arrows represent standard, high-level relationships between classes, where the direction of the arrow implies a dependency. Red arrows imply a dependency on time, i.e. we require timestamps in order to reassemble "slices" of the respective profiles.

[18] https://www.w3.org/TR/turtle/

The classes of the vCare ontology schema can be directly mapped to vCare profiles, where the clinical pathway class maps to the treatment plan profile. The state class represents a novel profile, which is essential for integrating time-dependent information needed for analysis and recommendation algorithms.

They define the required structure of the individual vCare profiles, such that it becomes explicit which information in which format has to be entered or generated. In addition to the structure, the identification of concepts and their instanced via URIs (i.e. unique resource identifiers) enables to link vCare knowledge to external ontologies in order to enable mutual enrichment.

To summarize the development of the vCare final version ontology: In **D4.2** we presented an high-level vCare ontology schema intentionally slightly simplifying the version presented in **D4.1** with the objective to balance complexity (with respect to the implementation for the Tech Lab phase) and interpretability (with respect to the core goal of the Tech Lab phase on gradually improving and finalizing the vCare ontology). The presented vCare schema overview modeled all vCare profiles, enabled to integrate all necessary data and information (e.g. clinical data, sensor data, pathways) and extended them with respect to information needed for advanced decision-making (e.g. for activity recommendation for pathways personalization). The essential differences (and reduction of complexity) comprise were:

- The re-naming of use case to coaching application (to be consistent with intended vCare profiles),
- The reduction of modelling states, observation features, agent profiles, policies and effects to a full state profile where all necessary information is modelled at a single point,
- The deferring of modelling topics for technical interaction among devices in the vCare architecture, as the Tech Lab phase will begin by a simpler, risk-aware setup to test the significant, essential decision-making capabilities based on expert rules,
- The modelling of games as specialization of activities, and
- The modelling of roles as specialization of user profiles.

2.3 Adjustments during the TechLab Phase with respect to D4.2

Based on the first version of the ontology, we made adjustments during the Tech lab phase to model the information we received from our partners during the integration more precisely.

The adaptations can be grouped into different parts:

- On the one hand, we have extended the ontology at various points by the information of the time, which is relevant for the collection of medical data of a patient, but also for the feedback of a result to an activity. In addition to time stamps, we have also added an attribute for modeling the current status at various points. So we can model for the clinical path of a patient whether it is an active plan, a first draft or if the plan was interrupted. We also modelled a similar status for all types of activities. By recording information on whether an activity was cancelled, rejected or completed, we can perform more complex analyses in the knowledge layer and better understand patterns of patient behavior.

- The second part of changes deals with the precise identification of information. In order to be able to precisely order the information transferred at the interfaces, we have introduced an ID for the patient as part of the integration, which enables a unique order in the mapping of the information.
- The third part with the adaptation deals with the mapping of a pathway and is related to the deliverables **D6.1**, **D6.2** and **D6.4** (see exemplary plan definition in **D6.4 annex 4**). As described here, we have added the attributes condition and relatedAction to the information about a clinical path. Perhaps complex relationships between different activities can be modeled. The differentiation between a general pathway for a specific disease and the designation of a tailor-made plan for a patient was addressed by the terms **plan definition** and **care plan**.
- We also introduced disease profiles for the remaining vCare disease types Heart failure, ischemic heart and Parkinson's.

The relation of the illustrated vCare ontology schema also directly maps to the medical ontology outline of **D1.3**, which we dwell on in Table 1. For the vCare ontology schema.

*Table 1: Comparison of vCare ontology schema and medical ontology outline of **D1.3***

vCare Ontology schema	*Medical Ontology Outline (D1.3)*	*Explanation*
User *(including Patient)*	Patient Profile (including Personal- and Clinical State), Roles	The patient profile class of the vCare ontology is a generalization of the patient profile of **D1.3**, as it also includes other profiles, such as for physicians. For each patient we track the clinical state over time and map the structure for the individual care plan.
Disease Pattern	Patient Profile – Clinical State	For the disease pattern profile, we have a direct mapping to parts of the clinical state of the patient of the medical ontology. The vCare ontology schema also comprises a designated clinical state profile, which includes links to the designated disease pattern profile.
Activity	Activities	Both ontologies comprise activities and are specialized by diagnosis-, treatment- and supporting activities. Based on the TechLab exchange of information about activities we extended the careplan holding information about

		activities to also hold information about related actions and conditions.
Device	Instrumental Parameters	The vCare ontology schema requires to model device profiles, where the sensor data is structured. Similarly, the medical ontology outline models instrumental parameters, which refer to the used devices as well.
Environment	Environmental Context	Both ontologies require classes for environments.
Plan Definition / Care Plan	Clinical Pathways	Both ontologies require classes for the clinical pathways. Within the vCare we further specify a generic Pathway as a Plan Definition and a tailored to the patient pathway as a Care Plan.
Coaching Application	Clinical Pathways	The vCare schema provides a designated profile for describing generalized use cases, where possible activities and patient profile classes are gathered. The profile is strongly related to Clinical Pathway templates (Plan Definition) and thus Clinical Pathways of the medical ontology.
Health Progress	Patient Profile	The health progress profile of the vCare schema gathers time-dependent versions of the patient profile in order to be able to assess the health progress in hindsight. It is thus related to the patient profile of the medical ontology.
State	-	The state profile is not directly modelled from the medical ontology perspective, as the state profile aggregates information for technical decision-making strategies

		(e.g. rule-based activity recommendation).
Evidence Indicators	Evidence Indicators, Assessment Documents	The vCare schema as well as the medical ontology comprise classes for evidence indicators, which support to quantify the success of recovery. We are mapping on the one hand the status of activities and the individual outcomes. On the other hand we are modelling the process of a patient's clinical state
Activity Schedule	Time Events	The time dependency of activity-dependent events is handled by the activity schedule class, which maps an activity to a time-period.
Feedbacks	Feedbacks	We have realised feedback on different levels. First we are adapting the recommended activities based on the results of the treatment- and diagnosis activities. Second we are tracking the status of activities (as described in the adjustments within the TechLab phase). Third, the vCare services provide questionnaires to investigate the patients wellbeeing. Last but not least we are getting feedback by including the expert rules in the reasoner. During the TechLab phase we have extended the rules and defined processes to conduct more complex rules.
(Expert rules in Notation3; collected in an individual repository)	Exceptional Procedures	The medical ontology models exceptional procedures, which support the safety of patients during activities. The latter are indirectly modelled for the vCare ontology: We exploit structured rules (modelled in Notation3; see Section 4) such that the Virtual Coach can engage in such reasoning.

3. Final Ontology Schema

In the following, we present the initial vCare ontology schema by describing and motivating the initial set of integrated ontologies, and by providing the actual RDF models of the vCare profiles.

We begin by giving an overview of the main structured vocabularies we use (see **Figure 3**). A vocabulary is always defined as *prefix* which defines the used namespace for the vCare ontology schema and links to the actual ontology on the Web. Next to the RDF vocabulary for modelling basic classes and properties, we make use of RDFS (i.e. Resource Description Framework Schema)[19] which extends RDF in order to, for example, model grouping of resources (e.g. by *rdfs:seeAlso* to link to semi-structured resources, such as standard websites). While we do not make use of OWL (i.e. Web Ontology Language)[20] reasoning, we use a limited set of its elements for the vCare ontology, such as expressing equality of resources via *owl:sameAs* (e.g. for diseases).

As mentioned, we make use SHACL (via the *sh* namespace) to model constraints to ensure that all required information is modelled with a controlled vocabulary. The XML schema (via the *xsd* namespace) supports modelling such constraints by defining several data types, such as *xsd:string*, to ensure correctness.

We, finally, define novel namespaces for the vCare schema (via the *vcs* namespace) as well as for vCare instances (via the *vci* namespace). The latter define vCare URIs (i.e. unique resource identifiers)[21] which allow to link information. While the vCare ontology is available as RDF file (e.g. in Turtle serialization) for the initial version (which suffices to load it into a Triple store to query required information), we plan to provide dereferenceable URIs for the final version of the vCare ontology, such that all resources can be directly queried.

[19] https://www.w3.org/TR/rdf-schema/

[20] https://www.w3.org/OWL/

[21] https://www.w3.org/TR/rdf-concepts/#section-Graph-URIref

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .


@prefix sh: <http://www.w3.org/ns/shacl#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix vcs: <http://ontology.vcare-
project.eu/vcare_ontology_schema.ttl#> .

@prefix vci: <http://ontology.vcare-
project.eu/vcare_ontology_instances.ttl#> .
```

Figure 3: Prefixes for vCare ontology (in Turtle syntax)

We now dwell on the individual schema details for each vCare profile.

3.1 Profiles

vCare profiles represent the main components of the vCare ontology and are modelled as RDF classes (*rdf:Class*), thus representing high-level concepts which can be directly instantiated or further refined as subclasses. In addition, vCare profiles are also typed as SHACL node shapes (*sh:NodeShape*), thereby enabling to model adequate constraints, such as required properties and/or their possible values. For the first version of the vCare ontology, we only make use of constraints for properties (*sh:property*) and constrain the values if needed. A label (*rdfs:label*) is always modelled in order to provide a textual representation for the profile.

Note that we often omit the model of the properties during the course of introducing the patterns, as they consist of simple structures with a human-interpretable label (via *rdfs:label*), the required subject (i.e. the resource from which the property originates, via *rdfs:domain)*, as well as the required object (i.e. the resource or literal the property leads to, via *rdfs:range*). The complete version of the vCare ontology is available under https://wisecloud.wiwi.tu-dresden.de/s/UVYqbbsIXpUjaYc (the vCare ontology schema and the vCare ontology instances).

Disease Pattern

Profiles for disease patterns are important for vCare ontology, as they model decisive information to distinguish among appropriate clinical care plan. The design of the profile is kept simple, as the focus lies on integrating knowledge from established ontologies and knowledge graphs. A disease is modelled as vCare concept and linked to existing representations via the *owl:sameAs* property, indicating that the concepts cover the same content. Throughout the

Tech lab phase, we extended the list of disease pattern profiles from Stroke to all vCare disease types, i.e. Stroke, Heart Failure, Heart Attack and Parkinson's (see use cases defined in **D1.2**). The set of integrated ontologies shown in **Figure 4**. These consist of LOINC (i.e. Logical Observation Identifiers Names and Codes)[22] for established concept identifiers for medical laboratory observations which are compatible with HL7 FHIR (i.e. Fast Healthcare Interoperability Resources; as also used as standard in **D6.1**, **D6.2**, **D6.3** and **D6.4**)[23], Human Disease Ontology[24] for established concepts of disease, RadLex[25] for already providing necessary concepts related to radiology, MeSH (i.e. Medical Subject Headings)[26] for a general, controlled medical vocabulary with links to central repositories such as PubMed[27], universAAL Health Measurement[28] for a rich, structured vocabulary for the vCare clinical state and Gene ontology[29] for providing knowledge on functions of genes. The initial set of medical ontologies were chosen based on their varying medical focus (i.e. the available concepts and relations complement each other) and their wide coverage of medical knowledge. While the current list of ontologies was carefully selected to enrich the vCare ontology, we will continue to integrate novel classes and instances throughout the Living lab phase, when novel relevant (medical-) information is resulting from its interactive tests.

Figure 4 shows the vCare Disease Pattern Profile, for which we first define novel namespaces for the medical ontologies. The disease profile is represented as a resource, as the vCare ontology schema namespace (*vcs*) is used. The semi-colon separators always indicate that novel property-object pairs follow, where the subject remains the same (i.e. all subsequently following properties following `vcs:DiseasePatternProfile` such as `rdf:type` correspond to the disease profile class). We therefore express that the disease profile is typed as `rdfs:Class` and `sh:NodeShape`. The former is a standard approach to model classes, whereas the latter indicates that the disease profile has to be checked by a SHACL reasoner. The `rdfs:label` simply defines a human-readable name for the RDF Class. The following `sh:property` entries use the concept of RDF blank nodes, which model new information (also referred to as RDF triples) without the need of resources. We first define that vCare ontology instances of the disease profile require a human-readable label (i.e. `rdfs:label`) using the `sh:path` property. All SHACL properties are assigned human-readable names (e.g. used for presenting error messages for reasoning) and can be extended with constraints, such as `sh:minCount` or `sh:maxCount` to make them mandatory. For the residual SHACL properties, we make use of existing RDF vocabularies to model textual descriptions (`rdfs:comment`), to model similarity of resources (`owl:sameAs`) and to model links to additional, possibly non-structured information (`rdfs:seeAlso`).

---

22 https://bioportal.bioontology.org/ontologies/LOINC
23 https://www.hl7.org/fhir/overview.html
24 https://bioportal.bioontology.org/ontologies/DOID
25 https://bioportal.bioontology.org/ontologies/RADLEX
26 https://bioportal.bioontology.org/ontologies/MESH
27 https://www.ncbi.nlm.nih.gov/pubmed/
28 http://ontology.universaal.org/HealthMeasurement.owl
29 http://geneontology.org/

```
@prefix loinc: <http://loinc.org/owl#> .

@prefix doid: <http://purl.obolibrary.org/obo/doid#> .

@prefix radlex: <http://radlex.org/RID/> .

@prefix mesh: <http://id.nlm.nih.gov/mesh/> .

@prefix gene: <http://identifiers.org/omim/> .

@prefix uvHealth: <http://ontology.universaal.org/HealthMeasurement.owl#>
.

vcs:DiseasePatternProfile ;

  rdf:type rdfs:Class ;

  rdf:type sh:NodeShape ;

  rdfs:label "vCare Disease Pattern Profile" ;

  sh:property [

      sh:path rdfs:label ;

      sh:name "name of disease" ;

      sh:minCount 1 ;

      sh:maxCount 1 ;

    ] ;

  sh:property [

      sh:path rdfs:comment ;

      sh:name "description of disease" ;

    ] ;

  sh:property [

      sh:path owl:sameAs ;

      sh:name "reference to ontology or knowledge graph" ;

    ] ;

  sh:property [

      sh:path rdfs:seeAlso ;

      sh:name "non- or semi-structured references for the disease" ;

    ] ;
```

```
.
```

Figure 4: Disease Pattern Profile for vCare ontology (in Turtle syntax)

User

User profiles are a central element of the vCare ontology, as they define required information about patients, but also important information about physicians such as their specialization. We model a user profile as RDF class with general information about the respective user, which is applicable to any user of the vCare platform (i.e. medical staff, technical staff as well as the patients). We therefore build on established ontologies for modelling user-specific information, as summarized in **Figure 5**, namely Friend-of-a-Friend (FOAF)[30] and vCard[31]. The latter are widely used for modelling general knowledge about users, focussing on demographic information.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
```

Figure 5: Prefixes for User Profile of vCare ontology (in Turtle syntax)

The general user profile class can then be defined in similar terms as the Person class of the FOAF ontology, for which we enforce the need to model a set of user details via SHACL properties with *sh:minCount 1*. Mandatory user details comprise the name, birthday, phone number, gender, marital status, family status, address and the job position. An email address remains optional. **Figure 6** shows an excerpt of the vCare user profile class. We exemplary limit the possible values for the marital status property by using *sh:in*, where only listed objects are allowed assignments.

```
vcs:UserProfile

  rdf:type rdfs:Class ;

  rdf:type sh:NodeShape ;

  rdfs:label "vCare User Profile" ;

  sh:property [

      sh:path foaf:givenName ;

      sh:maxCount 1 ;

      sh:minCount 1 ;
```

30 http://www.foaf-project.org/
31 https://www.w3.org/TR/vcard-rdf/

```
    sh:name "given name" ;
  ] ;
[…]
sh:property [
    sh:path vcs:maritalStatus ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:in (vcs:Married vcs:Single vcs:Divorced) ;
  ] ;
sh:property [
    sh:path vcard:hasEmail ;
    sh:name "email address" ;
  ] ;
.
```

Figure 6: User Profile of vCare ontology (in Turtle syntax)

On a schema level, we need to specialize user profiles for different vCare roles, such as patients or medical staff, where different information needs to be modelled in addition to the specified user profile details.

The Patient profile (shown in **Figure 7**) therefore specializes in the user profile through the subclass relationship *rdfs:subClassOf*, thus inheriting the modelled structure. As the patient profile comprises a personal- and a clinical state, we model them as separate classes, which are linked from the patient profile by properties *vcs:personalState* and *vcs:clinicalState*. In addition, the patient profile also links to an adequate clinical pathway profile, which will be presented subsequently. The property in *vcs:clinicalHistory* relates to the disease that the patient is suffering modeled by the diseaseProfile.

```
vcs:PatientProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:subClassOf vcs:UserProfile ;
  rdfs:label "vCare Patient Profile" ;
```

```
  sh:property [
      sh:path vcs:clinicalState ;
      sh:class vcs:ClinicalState ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
      sh:name "Clinical State" ;
    ] ;
  sh:property [
      sh:path vcs:clinicalHistory ;
      sh:class vcs:clinicalHistory ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
      sh:name "Clinical History" ;
    ] ;
  sh:property [
      sh:path vcs:personalState ;
      sh:class vcs:personalState ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
      sh:name "Personal State" ;
    ] ;
  sh:property [
      sh:path vcs:clinicalPathway ;
      sh:class vcs:ClinicalPathwayProfile ;
      sh:name "clinical pathway" ;
    ] ;
.
```

Figure 7: Patient Profile of vCare ontology (in Turtle syntax)

As the required information for both personal- and clinical state are strongly dependent on the medical use case (e.g. Stroke), we model the respective classes in their most general form and specialize them on a use-case basis (see Section 4 for the disease specialization). The respective states profiles are shown in **Figure 8**.

```
vcs:ClinicalState
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Clinical State" ;
.

vcs:PersonalState
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Personal State" ;
.
```

Figure 8: Clinical State of vCare ontology (in Turtle syntax)

Activity

vCare profiles for activities define possible approaches to support rehabilitation and strongly depend on the given use case (i.e. the coaching application). As the activities might originate based on scientific studies, we enable references to external documents describing the activity in terms of procedure but also positive empirical evidence. Even if a scientific study is not available, it is important to have additional information about the activity in a human-readable format. This is achieved by enforcing to model at least one value for property *rdfs:seeAlso*, where the value is a hyperlink (using *sh:minCount*). The same condition should hold for thorough textual descriptions of the activity, which is modelled via *rdfs:comment*. We, finally, want to relate activities with evidence indicators, such that we can directly process the results. **Figure 9** presents the resulting activity profile.

```
vcs:ActivityProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Activity" ;
```

```
  sh:property [
      sh:path rdfs:seeAlso ;
      sh:minCount 1 ;
      sh:name "further references for activity" ;
    ] ;
  sh:property [
      sh:path rdfs:comment ;
      sh:minCount 1 ;
      sh:name "textual description of activity" ;
    ] ;
  sh:property [
      sh:path vcs:evidenceIndicator ;
      sh:class vcs:EvidenceIndicator ;
      sh:name "evaluation metric to use for activity" ;
    ] ;
sh:property [
      sh:path vcs:status ;
      sh:class vcs:status ;
      sh:name "status of the activity" ;
    ] ;
.
```

Figure 9: Activity Profile of vCare ontology (in Turtle syntax)

Evidence indicators are modelled as separate classes with a rather generic structure. Besides the option of linking to an existing ontology where devices etc. are modelled, we include a textual description and the results.

```
vcs:EvidenceIndicatorProfile
  rdf:type rdfs:Class ;
```

```
  rdf:type sh:NodeShape ;
  rdfs:label "vCare evidence indicator" ;
  sh:property [
      sh:path rdfs:seeAlso ;
      sh:name "further references for evidence indicator" ;
    ] ;
  sh:property [
      sh:path rdfs:comment ;
      sh:minCount 1 ;
      sh:name "textual description of evidence indicator" ;
    ] ;
sh:property [
      sh:path vcs:status ;
      sh:class vcs:result ;
      sh:name "result documentation" ;
    ] ;
.
```

Figure 10: Evidence Indicator Profile of vCare ontology (in Turtle syntax)

We partition activities into diagnosis activities, supporting activities and treatment activities, where the former two activities only observe behaviour to calculate evidence indicators and the latter involves actual participation of the patient. All are specializations of the activity profile. Diagnosis- and supporting activities require to model a device (see Paragraph about Device profiles) which supports to measure the activity, as well as the frequency of evaluation (e.g. daily). We show an example of a footstep counter for Stroke rehabilitation in Section 4.

For diagnosis activities, we only require to model the involved supervision during the activity, which might be by an actual physician or the Virtual Coach. **Figure 11** summarizes the classes for diagnosis- and treatment activities (note that supporting activities are modelled equally to diagnosis activities).

```
vcs:DiagnosisActivity
  rdf:type rdfs:Class ;
```

```
  rdf:type sh:NodeShape ;
  rdfs:subClassOf vcs:ActivityProfile;
  sh:property [
      sh:path vcs:device ;
      sh:name "measuring device for diagnosis activity" ;
    ] ;
  sh:property [
      sh:path vcs:evaluationFrequency ;
      sh:name "frequency for evaluation gathered information" ;
    ] ;
.

vcs:TreatmentActivity
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:subClassOf vcs:ActivityProfile;
  sh:property [
      sh:path vcs:supervisedBy ;
      sh:name "supervised by" ;
    ] ;
.
```

Figure 11: Specialized Activities of vCare ontology (in Turtle syntax)

Treatment Plan

A vCare treatment plan models clinical pathways as defined in **D6.1**, **D6.2**, **D6.3**, **D6.4**, consisting of scheduled activities. The model is suitable for the Tech Lab phase, as the goal is to gradually target increasingly complex use cases, starting with a single activity. Note that generalized pathway templates can be modelled as collection of activity profiles in the same way.

We first model scheduled activities (see **Figure 12**) by defining an upper class, linking to an activity (or a specialization thereof), assigning start- and end dates, as well as the results of the activity in the form of the calculated evidence indicators.

```
vcs:ActivitySchedule
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Activity Schedule" ;
  sh:property [
      sh:path vcs:activityProfile ;
      sh:name "activity" ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
      sh:class vcs:Activity ;
    ] ;
  sh:property [
      sh:path vcs:activeDevice ;
      sh:name "active device" ;
      sh:class vcs:DeviceProfile ;
    ] ;
  sh:property [
      sh:path vcs:startTime ;
      sh:name "start time of activity" ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
    ] ;
  sh:property [
      sh:path vcs:endTime ;
      sh:name "end time of activity" ;
      sh:maxCount 1 ;
```

```
      sh:minCount 1 ;
    ] ;
  sh:property [
      sh:path vcs:goal ;
      sh:name "goal of activity" ;
    ] ;
  sh:property [
      sh:path vcs:result ;
      sh:name "result of activity" ;
    ] ;
  sh:property [
      sh:path vcs:documentedSupervision ;
      sh:name "documented supervision" ;
    ] ;

sh:property [
      sh:path vcs:relatedAction ;
      sh:name "reference to related action" ;
    ] ;

sh:property [
      sh:path vcs:condition ;
      sh:name "condition to the related action" ;
    ] ;
.
```

Figure 12: Activity Scheduler of vCare ontology (in Turtle syntax)

A clinical pathway (modelled in **Figure 13**), representing the treatment profile, can then be simply modelled as collection of scheduled clinical pathways, which is assigned to a patient.

```
vcs:ClinicalPathwayProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Clinical Pathway Profile" ;
  sh:property [
      sh:path vcs:scheduledActivity ;
      sh:name "scheduled Activity" ;
      sh:class vcs:AcitvitySchedule ;
    ] ;

sh:property [
      sh:path vcs:carePlanDescription ;
      sh:name "description of the care plan" ;
      sh:class vcs:carePlanDescription ;
    ] ;
sh:property [
      sh:path vcs:carePlanStatus ;
      sh:name "status of the care plan" ;
      sh:class vcs:carePlanStatus ;
    ] ;
.
```

Figure 13: Clinical Pathway Profile of vCare ontology (in Turtle syntax)

Device

Profiles for devices are crucial for integrating the heterogeneous landscape of sensors, which are used to support patient rehabilitation applications. The vCare device profile, as presented in **Figure 13**, is modelled by building on the established Semantic Sensor Network Ontology[32] and its lightweight version SOSA (abbreviation for Sensor, Observation, Sample, and Actuator)[33]. SSN and SOSA are widely adopted for modelling Internet-of-Things (IoT) devices

[32] https://www.w3.org/TR/vocab-ssn/

[33] Haller, A., Janowicz, K., Cox, S.J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R. and Stadler, C., 2018. The SOSA/SSN ontology: a joint WeC and OGC standard specifying the semantics of sensors observations actuation and sampling. *Semantic Web*, *1*, pp.1-19.

and –sensors, which provide an eligible baseline for modelling devices presented in **D7.4**, which can then be linked to medical ontologies such as MeSH.

It is sensible to model device profiles as specialization of *sosa:Platform*, representing platforms which host sensors. We can then enforce that the SOSA property *sosa:hosts* has to be modelled, leading to available sensors on a particular device. In order to gather a catalogue of general devices, we model a device class where we can collect available device types irrespectively of the actual physical devices.

```
@prefix sosa: <http://www.w3.org/ns/sosa/> .


vcs:Device
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Device" ;
  sh:property [
      sh:path rdfs:label ;
      sh:name "name of device" ;
    ] ;
  sh:property [
      sh:path rdf:type ;
      sh:name "type of device" ;
    ] ;
  sh:property [
      sh:path rdfs:seeAlso ;
      sh:name "link to further information about vendor or product" ;
    ] ;
.


vcs:DeviceProfile ;
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
```

```
  rdfs:label "vCare Device Profile" ;
  owl:sameAs sosa:Platform ;
  sh:property [
      sh:path rdfs:label ;
      sh:name "name of device" ;
    ] ;
  sh:property [
      sh:path vcs:device ;
      sh:name "type of device" ;
    ] ;
  sh:property [
      sh:path sosa:hosts ;
      sh:name "hosted sensor" ;
    ] ;
.
```

Figure 14: Device Profile of vCare ontology (in Turtle syntax)

Environment

In order to estimate the current location of the patient during her/his daily routine, we need to specify an environment profile. The latter is linked to available devices, which support estimating the current position of the patient. The required structure of the vCare ontology class for environment profiles shown in **Figure 15** is straightforward, in that we need structured representation of the environment itself (modelled via a separate environment class, accessed via *vcs:patientEnvironment*) with limited possible values (limited to actual available environments), a current timestamp to persist where the patient was observed at which time, as well as the sensor which was used for locating (modelled via *sosa:hosts*).

```
vcs:EnvironmentProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Environment Profile" ;
```

```
  sh:property [
      sh:path vcs:patientEnvironment ;
      sh:name "current patient environment" ;
      sh:class vcs:Environment ;
      sh:in (vci:Indoors vci:Outdoors) ;
    ] ;
  sh:property [
      sh:path vcs:timeStamp ;
      sh:name "time stamp" ;
    ] ;
  sh:property [
      sh:path sosa:hosts ;
      sh:name "sensor from which the environment was detected" ;
    ] ;
.
```

Figure 15: Environment Profile of vCare ontology (in Turtle syntax)

Coaching application

The coaching application profile is strongly related to use case description, as provided in **D1.2**. It describes the goal of the rehabilitation process for a class of patients. Based on the available use cases, we define an initial version of the coaching application profile in **Figure 15**, consisting of a textual description, a set of possible activities, a link to a structured representation of the involved disease (e.g. Stroke) as well as a link to eligible patient profile classes (e.g. a patient profile for Strokes, as introduced in **Section 4**).

```
vcs:CoachingApplicationProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Coaching Application Profile" ;
  sh:property [
```

```
    sh:path rdfs:label ;
    sh:name "name of coaching application" ;
    sh:minCount 1 ;
  ] ;
sh:property [
    sh:path rdfs:comment ;
    sh:name "textual description of coaching application" ;
    sh:minCount 1 ;
  ] ;
sh:property [
    sh:path vcs:activity ;
    sh:name "possible activities for the coaching application" ;
    sh:minCount 1 ;
  ] ;
sh:property [
    sh:path vcs:disease ;
    sh:name "The underliyng disease for coaching application" ;
    sh:minCount 1 ;
  ] ;
sh:property [
    sh:path vcs:personalStateProfile ;
    sh:name "reference to eligible personal state class" ;
    sh:minCount 1 ;
  ] ;
sh:property [
    sh:path vcs:clinicalStateProfile ;
    sh:name "reference to eligible clinical state class" ;
    sh:minCount 1 ;
  ] ;
```

```
.
```

Figure 16: Coaching Application Profile of vCare ontology (in Turtle syntax)

Health Progress

The profile for a patient's health progress needs to access a history of patient profiles at different time steps of the rehabilitation process. For the initial version of the vCare ontology, we model a minimal version of the health progress profile by only collecting patient profiles at time different time steps. The latter can then be analysed based on different instances of *vcs: TimeDependentPatientProfile*. The latter is a wrapper class (similar to ActivitySchedule), which persists a Patient profile instance for a particular time stamp, as the latter changes over time (i.e. in terms of the patient's clinical state). **Figure 18** shows the respective RDF Class of the health progress profile.

```
vcs:<clinical parameter>
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Health Progress Profile of <clinical parameter>" ;
 sh:property [
      sh:path vcs:<clinical parameter value> ;
      sh:name "value of the clinical parameter observed" ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
    ] ;
sh:property [
      sh:path vcs:timeStamp ;
      sh:name "time stamp referring to the time the value was observed" ;
      sh:minCount 1 ;
    ] ;
.
```

Figure 17: Clinical Parameter Profile of vCare ontology (in Turtle syntax)

```
vcs:HealthProgressProfile
```

```
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Health Progress Profile" ;
  sh:property [
      sh:path rdfs:timeDependentPatientProfile ;
      sh:name "a wrapper of the patient profile at a particular time." ;
      sh:minCount 1 ;
      sh:class vcs:TimeDependentPatientProfile
    ] ;
.
```

Figure 18: Health Progress Profile of vCare ontology (in Turtle syntax)

3.2 Decision-Theoretic Model

Based on the presented vCare profiles, we can model a *meta-profile* for decision-theoretic recommendation tasks of the Virtual Coach. The resulting profile needs to capture all required information to make use of rule-based algorithms, Machine Learning-based algorithms, as well as Reinforcement Learning-based algorithms, as outlined in **D4.1**.

The State profile shown in **Figure 19** integrates information about the current patient profile, the current environment profile as well as the current clinical pathway. As such, the state profile models all information which are used as input for the reasoning tasks, e.g. to decide how to adapt the clinical pathway to foster personalization for patients.

```
vcs:StateProfile
  rdf:type rdfs:Class ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare State Profile" ;
  sh:property [
      sh:path vcs:patientProfile ;
      sh:name "reference to current patient profile" ;
      sh:class vcs:PatientProfile ;
```

```
      sh:maxCount 1;
      sh:minCount 1 ;
    ] ;
  sh:property [
      sh:path vcs:environmentProfile ;
      sh:name "reference to current environment profile" ;
      sh:class vcs:EnvironmentProfile ;
      sh:maxCount 1;
      sh:minCount 1 ;
    ] ;
  sh:property [
      sh:path vcs:clinicalPathway ;
      sh:name "reference to current clinical pathway profile" ;
      sh:class vcs:ClinicalPathwayProfile ;
      sh:maxCount 1;
      sh:minCount 1 ;
    ] ;
.
```

Figure 19: State Profile of vCare ontology (in Turtle syntax)

Given the final vCare schema as presented in **Section 3**, we now present vCare ontology instances for the exemplary use case of different disease rehabilitation. We design the instances based on the use case description mapped to a structured knowledge representation, as available in **D1.3** and extend the information included based on the adjustments during the TechLab phase. Note that we are not modelling a particular patient, but focussing on enabling to model any patient for a minimal set of relevant clinical information. The residual profiles, i.e. treatment plan, environment, health progress and state, the instances are direct extensions of the presented profiles with time stamps (in order to analyse and recommend).

4. Exemplary Patient Profiles

In the following we will introduce exemplary patient profiles for the different disease cases adresse in the vCare project. Based on the example of patients, as introduced in deliverables

D.1.2, **D1.3**, **D1.4**, we present exemplary vCare ontology instances for each of the pathology. The profiles are structured into disease pattern, patient, activity, device and coaching application.

4.1 Disease Pattern Profile

We now introduce disease pattern profiles for each vCare disease type by modelling respective subclasses. Based on the schema definition, we focus on linking the disease to available ontologies. We used, if available, concepts from Radlex, Human Disease Ontology, Gene Ontology, LOINC and Medical Subject Headings, but want to mention that the latter provide further references to other ontologies, which can be directly queried. These integrated ontologies suffice to define the vCare disease types in order to potentially link to other resources. The used linked concepts were evaluated by physicians and can be used to collect respective disease taxonomies or possible treatments as reported in mentioned studies.

In the following, we will discuss the different ontology models of the underlying disease profiles. The general profiles serve as a basis for the highly personalized and individualized recording of the individual disease pattern. The modelled disease profiles are linked to existing widespread ontologies, which can be used for further knowledge aggregation.

The Disease Pattern for patients with heart failure is shown in **Figure 20.**

Disease Pattern - Heart Failure

```
vcs:HeartFailureProfile
  rdf:type sh:NodeShape ;
  rdfs:subClassOf vcs:DiseasePatternProfile ;
  owl:sameAs radlex:RID34795 ;
  owl:sameAs mesh:D006333 ;
  owl:sameAs loinc:LA24542-5 ;
  rdfs:label "vCare Heart Failure Disease Pattern Profile" ;
.
```

Figure 20: Heart Failure Disease Pattern Profile of vCare ontology (in Turtle syntax)

In the same way the generic baseline for the vCare ontology is modelled for the ischemic heart disease (see **Figure 21**).

Disease Pattern - Ischemic Heart

```
vcs:IschemicHeartProfile
  rdf:type sh:NodeShape ;
```

```
  rdfs:subClassOf vcs:DiseasePatternProfile ;
  owl:sameAs radlex:RID3235 ;
  owl:sameAs  mesh:D017202 ;
  owl:sameAs loinc:78976-8 ;
  rdfs:label "vCare Ischemic Heart Disease Pattern Profile" ;
.
```

Figure 21: Ischemic Heart Disease Pattern Profile of vCare ontology (in Turtle syntax)

As the last of the diseases addressed in vCare, we consider the modelling of parkinson's disease, as shown in **Figure 22**.

Disease Pattern - Parkinson

```
vcs:ParkinsonProfile
  rdf:type sh:NodeShape ;
  rdfs:subClassOf vcs:DiseasePatternProfile ;
  owl:sameAs radlex:RID5163 ;
  owl:sameAs gene:616361 ;
  owl:sameAs loinc:45660-8 ;
  rdfs:label "vCare Parkinson Disease Pattern Profile" ;
.
```

Figure 22: Parkinson Disease Pattern Profile of vCare ontology (in Turtle syntax)

We finally deal with disease type for which we model a Stroke hierarchy based on the Stroke Ontology[34], which only allows instances of types “Intracranial haemorrhage” and “Ischemic stroke”. The resulting Stroke disease pattern profile can then be instantiated for the individual types in order to link to further ontologies. The Stroke disease pattern profile is presented in **Figure 23**.

Disease Pattern - Stroke

```
vcs:StrokeProfile
  rdf:type sh:NodeShape ;
```

[34] https://bioportal.bioontology.org/ontologies/STO

```
  rdfs:subClassOf vcs:DiseasePatternProfile ;
  owl:sameAs sto:Stroke_type ;
  rdfs:label "vCare Stroke Disease Pattern Profile" ;
  sh:property [
      sh:path vcs:stroke_type ;
      sh:name "type of Stroke" ;
      sh:in (sto:Intracranial_hemorrhage sto:Ischemic_stroke ) ;
      sh:minCount 1;
      sh:maxCount 1;
    ] ;
.
```

Figure 23: Stroke Disease Pattern Profile of vCare ontology (in Turtle syntax)

Figure 24 exemplarily shows both the vCare ontology instance for Stroke disease, which is an instance of the vCare Stroke disease pattern profile. The *owl:sameAs* property is used to point to the same concept in other ontologies. The individual concepts are abbreviated by ontology-dependent identifiers, for which individual resources are available in the respective ontologies.

```
vci:Stroke
  rdf:type vcs:StrokeProfile ;
  rdf:label "Stroke" ;
  owl:sameAs radlex:RID5178 ;
  owl:sameAs doid:6713 ;
  owl:sameAs gene:601367 ;
  owl:sameAs mesh:D020521 ;
  owl:sameAs loinc:54807-3;
.
```

Figure 24: Stroke instance of vCare ontology (in Turtle syntax)

4.2 Patient Profile

We proceed by modelling the disease specific patient profile that include the clinical state and personal state. the clinical condition is modelled in a disease-specific way as defined in **D1.3**. Similarly, we present personal profiles for the individual diseases with general and disease-specific personal attributes. During the consultations with the medical partners for the derivation of further expert rules, the attributes defined here were sufficient to characterise and analyse the condition. However, the practicability will be proven in the Living lab. Accordingly, the modeling of the personal condition can be understood as a living document.

Clinical State- Stroke

For the clinical state of stroke patients (see **Figure 25**), we exemplarily define several pieces of relevant information, such as range of motion or the result of the ARAT test (short for Action Research Arm Test), as defined in **D1.3** for Stroke patients. We additionally add generally relevant clinical information, such as the body weight or the heart rate, which are available via the universAAL Health Measurement ontology.

```
vcs:StrokeClinicalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:ClinicalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Stroke Clinical State" ;
  sh:property [
      sh:path vcs:motionRange ;
      sh:name "range of motion" ;
    ] ;
  sh:property [
      sh:path vcs:miniBESTest ;
      sh:name "functional limitations in gait and balance" ;
    ] ;
  sh:property [
      sh:path vcs:aratTest ;
      sh:name "reduction of upper limb dexterity" ;
    ] ;
 sh:property [
      sh:path vcs:vasTest ;
```

```
      sh:name "test on patients pain level" ;
    ] ;
  sh:property [
      sh:path vcs:borgTest ;
      sh:name "score of patient in borg assessment" ;
    ] ;
   sh:property [
      sh:path vcs:swallowingTroubles ;
      sh:name "troubles for swallowing" ;
    ] ;
   sh:property [
      sh:path vcs:fatigue ;
      sh:name "getting tired quickly (fatigue)" ;
    ] ;
   sh:property [
      sh:path vcs:bodyWeight ;
      sh:name "body weight of patient" ;
      sh:class uvHealth:PersonWeight ;
    ] ;
   sh:property [
      sh:path vcs:heartRate ;
      sh:name "heart rate of patient" ;
      sh:class uvHealth:HeartRate ;
    ] ;
   sh:property [
      sh:path vcs:numberSteps ;
      sh:name "number of steps the patient performed that day" ;
    ] ;
   sh:property [
```

```
    sh:path vcs:thresholdNumberSteps ;
    sh:name "threshold number of steps the patient should perform" ;
  ] ;
  sh:property [
    sh:path vcs:timeInactivity ;
    sh:name "time the patient has been inactive" ;
  ] ;
  sh:property [
    sh:path vcs:thresholdTimeInactivity ;
    sh:name "threshold time of inactivity" ;
  ] ;
.
```

Figure 25: Stroke Clinical State of vCare ontology (in Turtle syntax)

Clinical State- Heart Failure

For the clinical state of heart failure patients (see **Figure 26**), we include information about the patient caloric intake, oedemas, angina and other attributes as defined in **D1.3** for heart failure patients in their clinical state.

```
vcs:HeartFailureClinicalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:ClinicalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Heart Failure Clinical State" ;
  sh:property [
    sh:path vcs:bmi ;
    sh:name "body mass index" ;
  ] ;
  sh:property [
    sh:path vcs:targetCaloric ;
```

```
        sh:name "target caloric intake for the patient" ;
    ] ;
  sh:property [
        sh:path vcs:palpitations ;
        sh:name "information about heart palpitations" ;
    ] ;
  sh:property [
        sh:path vcs:oedemas ;
        sh:name "oedemas" ;
    ] ;
  sh:property [
        sh:path vcs:angina ;
        sh:name "describing patients psychological angina suffering" ;
    ] ;
  sh:property [
        sh:path vcs:dizziness ;
        sh:name "information about dizziness" ;
    ] ;
  sh:property [
        sh:path vcs:dyspnoea ;
        sh:name "describing patients shortness of breath" ;
    ] ;
 sh:property [
        sh:path vcs:cardiopulmonary ;
        sh:name "cardiopumonary fitness" ;
    ] ;
 sh:property [
        sh:path vcs:neuromuscularIntegrity ;
        sh:name "neuromuscular integrity" ;
```

```
    ] ;
.
```

Figure 26: Heart Failure Clinical State of vCare ontology (in Turtle syntax)

Clinical State- Ischemic Heart Disease

For the clinical state of ischemic heart disease patients (see **Figure 25**), we include information about the patient hip and waist circumference and other attributes as defined in **D1.3** for ischemic heart disease patients in their clinical state.

```
vcs:IschemicHeartClinicalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:ClinicalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Ischemic Heart Disease Clinical State" ;
  sh:property [
      sh:path vcs:bmi ;
      sh:name "body mass index" ;
    ] ;
sh:property [
      sh:path vcs:waistCircumference ;
      sh:name "waist circumference" ;
    ] ;
sh:property [
      sh:path vcs:hipCircumference ;
      sh:name "hip circumference" ;
    ] ;
  sh:property [
      sh:path vcs:targetCaloric ;
      sh:name "target caloric intake for the patient" ;
```

```
    ] ;
 sh:property [
      sh:path vcs:angina ;
      sh:name "describing patients psychological angina suffering" ;
    ] ;
  sh:property [
      sh:path vcs:dyspnoea ;
      sh:name "describing patients shortness of breath" ;
    ] ;
 sh:property [
      sh:path vcs:cardiopulmonary ;
      sh:name "cardiopumonary fitness" ;
    ] ;
sh:property [
      sh:path vcs:palpitations ;
      sh:name "information about heart palpitations" ;
    ] ;
.
```

Figure 27: Ischemic Heart disease Clinical State of vCare ontology (in Turtle syntax)

Clinical State-Parkinson

For the clinical state of parkinson patients (see **Figure 26**), we include the following information as defined in **D1.3** for parkinson patients in their clinical state.

```
vcs:ParkinsonClinicalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:ClinicalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Parkinson Disease Clinical State" ;
```

```
  sh:property [
      sh:path vcs:motorStatus ;
      sh:name "information about patients ability of body movement" ;
    ] ;
sh:property [
      sh:path vcs:cognitiveStatus ;
      sh:name "nested information about cognitiv wellbeeing" ;
    ] ;
sh:property [
      sh:path vcs:neuropsychiatricStatus ;
      sh:name "summary of the neuropsychiatric attributes" ;
    ] ;
  sh:property [
      sh:path vcs:orthostaticHypotension ;
      sh:name "information about orthostatic hypotensions" ;
    ] ;
 sh:property [
      sh:path vcs:angina ;
      sh:name "describing patients psychological angina suffering" ;
    ] ;
  sh:property [
      sh:path vcs:sleepDisorder ;
      sh:name "patient suffering sleep disorders" ;
    ] ;
.
```

Figure 28: Parkinson disease Clinical State of vCare ontology (in Turtle syntax)

<u>Personal State</u>

In a similar vein, we model the personal state of the patients. In the following we first introduce the personal state of Stroke patients in **Figure 27** with information about the current mood of the patient (e.g. limited to several pre-defined mood states) or her/his current appearance. The modelling of personal state for heart failure and ischemic heart disease cases are further extensions to the general personal state (see **Figure 28**).

```
vcs:StrokePersonalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:PersonalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Stroke Personal State" ;
  sh:property [
      sh:path vcs:appearance ;
      sh:name "appearance" ;
    ] ;
  sh:property [
      sh:path vcs:attitude ;
      sh:name "attitude" ;
    ] ;
  sh:property [
      sh:path vcs:affect ;
      sh:name "affect" ;
    ] ;
  sh:property [
      sh:path vcs:mood ;
      sh:name "mood" ;
      sh:in (vcs:anxious vcs:depressed vcs:dysphoric vcs:euphoric
vcs:irritated) ;
    ] ;
  sh:property [
      sh:path vcs:adl ;
      sh:name "activity of daily living measured based on index of
independence scale" ;
    ] ;
```

```
.
```

Figure 29: Stroke Personal State of vCare ontology (in Turtle syntax)

Following the turtle syntax for modelling the personal state of heart failure, parkinson and ischemic heart disease cases are provided (**Figure 30**).

```
vcs:IschemicHeartFailurePersonalState
  rdf:type rdfs:Class ;
  rdfs:subClassOf vcs:PersonalState ;
  rdf:type sh:NodeShape ;
  rdfs:label "vCare Ischemic or Heart Failure Personal State" ;
  sh:property [
      sh:path vcs:smokingStatus ;
      sh:name "status on patients smoking habits" ;
    ] ;
   sh:property [
      sh:path vcs:alcoholConsumption ;
      sh:name "status on patients alcohol consumption habits" ;
    ] ;
   sh:property [
      sh:path vcs:eatingHabits ;
      sh:name "status on patients eating habits" ;
    ] ;
  sh:property [
      sh:path vcs:appearance ;
      sh:name "appearance" ;
    ] ;
  sh:property [
      sh:path vcs:attitude ;
      sh:name "attitude" ;
```

```
    ] ;
  sh:property [
      sh:path vcs:affect ;
      sh:name "affect" ;
    ] ;
  sh:property [
      sh:path vcs:mood ;
      sh:name "mood" ;
      sh:in (vcs:anxious vcs:depressed vcs:dysphoric vcs:euphoric
vcs:irritated) ;
    ] ;
  sh:property [
      sh:path vcs:adl ;
      sh:name "activity of daily living measured based on index of
independence scale" ;
    ] ;
.
```

Figure 30: Heart Failure, Parkinson and Ischemic Heart Disease Personal State of vCare ontology (in Turtle syntax)

Activity

As defined in **D1.3** (see page 19), a possible diagnosis activity for Stroke rehabilitation is to observe motor status updates, which can be retrieved via in-home- and wearable motion sensors. We therefore model the *motor status updates* diagnosis activity, which is assigned the evidence indicator *number of steps*, which measures the *daily* taken steps of the patient via a Fitbt Charge 3 *device*. The respective vCare ontology instance is shown in **Figure 31**. Other Stroke activities (also for treatment or support) are modelled in the same way.

```
vci:MotorStatusUpdates
  rdf:type vcs:DiagnosisActivity ;
  rdfs:comment "motor status observation" ;
  vcs:evidenceIndicator vci:NumberOfSteps ;
  vcs:evaluationFrequency vci:Daily ;
```

```
  vcs:device vci:Charge3 ;
.
```

Figure 31: Footstep Activity Counter Activity of vCare ontology (in Turtle syntax)

Device

For the example of counting footsteps as diagnosis activity, we model an exemplary shake sensor in **Figure 32**, which is integrated into a Fitbit Charge 3 device, as defined in **D7.4**. We integrate different domain ontologies to enrich the vCare ontology. We use the Product Ontology[35] to integrate information about Fitbit, M3 lite (i.e. Machine-to-Machine Measurement)[36] to integrate information about shake sensors (available as instance) and SOSA to model the availability of sensors (as defined in the vCare ontology schema). The product ontology provides valuable knowledge about a variety of available products which can be directly used for devices presented in **D7.4**. M3 lite, in addition, comprises structured information about a variety of sensors (as well as their required inputs, outputs, and respective formats), which are required for integrating the devices in the vCare ontology.

```
@prefix m3lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix pto: <http://www.productontology.org/id/> .

vci:Charge3
  rdf:type vcs:Device ;
  rdf:type pto:Fitbit ;
  rdfs:label "physical device" ;
  rdfs:seeAlso <https://www.fitbit.com/eu/charge3> ;

vci:ShakeSensor_1 a m3lite:ShakeSensor .

vci:Charge3_1
  rdf:type vcs:DeviceProfile ;
  rdfs:label "Charge 3 with id 1" ;
```

35 http://www.productontology.org/
36 https://lov.linkeddata.es/dataset/lov/vocabs/m3lite

```
  vcs:device vci:Charge3 ;

  sosa:hosts vci:ShakeSensor_1 ;

.
```

Figure 32: Charge3 Device of vCare ontology (in Turtle syntax)

Coaching Application

An example for a Stroke coaching application profile (see **Figure 33**) contains a textual description of the problem (extracted from **D1.2**), references to available vCare activities which are applicable, as well as personal- and clinical state profiles, which can be directly used for novel Stroke patients.

```
vci:StrokeCoachingApplication

  rdf:type vcs:CoachingApplicationProfile ;

  rdfs:label "Stroke Coaching Application" ;

  rdfs:comment "Coaching application for patients who suffered a left-
sided cerebral stroke that caused a right hemiparesis. This deficit
causes functional motor impairment and problems managing life both in
domestic environments and in open spaces." ;

  vcs:activity vci:EmotionalSocialTherapy, vci:MemoryTraining,
vci:SeriousGame, vci:MotorStatusUpdates ;

  vcs:personalStateProfile vcs:StrokePersonalState ;

  vcs:clinicalStateProfile vcs:StrokeClinicalState ;

.
```

Figure 33: Stroke Coaching Application of vCare ontology (in Turtle syntax)

EXPERT RULES

We show how to model expert rules for the exemplary Stroke use case. Expert rules are a valuable means to include strong medical domain knowledge into the vCare ontology as well as into the automatic adaption capabilities (e.g. with respect to activities in a clinical care plan). We present two valuable scenarios, namely (1) the inference of novel information from medical evaluations (such as the Mini-BESTTest; see **D1.3** page 17) which enables interpretability and is highly useful for algorithms for automatic adaption, and (2) the rule-based adaptation of activities based on the resulting inferred results of medical evaluations, which is highly useful for clinical care plan personalization. For all applications of expert rules, we make use of

Notation3 (N3)[37], as discussed in Section 3.1, and thoroughly motivated and introduced in **D4.1**. The set of expert rules are stored separately to the vCare ontology.

Inference of novel information

Expert rules can be directly used for the inference of novel facts for the vCare ontology, i.e. novel vCare ontology instances. We show the mechanism based on the example of inferring the degree of limitations of a patient based on Mini-BESTest. The rule, as depicted in **D1.3** (page 17), is the following:

- Severe limitations **if** Mini-BESTest <= 18
- Mild limitations **if** Mini-BESTest >= 19

The goal of inference of novel information is thus to automatically derive the qualitative evaluation of the Mini- BESTest results. Given that we have instances of patient profiles for patients, we define the following general rule, expressing that valid patient profiles for Stroke, where the clinical state is typed accordingly, should be evaluated for the Mini-BESTest value. If the value is smaller or equal to 18, we trigger the generation of a novel piece of information, which is appended to the clinical state of the patient. The resulting rule is depicted in **Figure 34**. The same rule, for values greater or equal to 19 can be defined accordingly.

```
{
    ?patient_profile a vcs:PatientProfile .
    ?patient_profile vcs:clinicalState ?clinical_state .
    ?clinical_state rdf:type vcs:StrokeClinicalState .
    ?clinical_state vcs:miniBESTest ?mini_bestest .
    ?mini_bestest <= 18 .
}
=>
{
    ?clinical_state vcs:limitations vci:mild .
}.
```

Figure 34: Notation3 rule for inferring limitation states

[37] https://www.w3.org/TeamSubmission/n3/

The resulting information of, for example, a mild limitation can then be used as qualitative means for interpretability (e.g. for being able to trace back how the assessment of a *mild* observation was made) or for direct use in activity adaption, where the novel information is used to adapt the clinical pathway of a patient.

Activity adaptation

Other than information inference, or goal is now to adapt the scheduled activity for a patient. A possible expert rule could express that patients with *mild limitations* (as inferred via the results of the Mini-BESTest) and successful achievement of an activity goal (here exemplarily set to number of footsteps per week) are assigned adapted activities with increased goals (here an increased number of footsteps), i.e.:

- Increase goal with respect to footsteps **if** mild limitations **and** achieved footsteps > old goal

An exemplary expert rule adjusting the goal value for a specific activity is presented in **Figure 35**, where the variable *?new_goal* has to be inferred or manually set. One could, for example, infer to novel value by simply adding a predefined number of steps (e.g. 200) to the old goal or adaptively set the novel value via Machine Learning.

```
{
    ?patient_profile a vcs:PatientProfile .
    ?patient_profile vcs:clinicalState ?clinical_state .
    ?clinical_state vcs:limitations vci:mild .

    ?patient_profile vcs:clinicalPathway ?clinicalPathway .
    ?clinicalPathway vcs: scheduledActivity ?scheduled_activity .
    ?scheduled_activity vcs:activity vci:MotorStatusUpdates .
    ?scheduled_activity vcs:result ?result .
    ?scheduled_activity vcs:goal ?old_goal .
    ?result >= ?old_goal .
}
=>
{
    vci:patientActivity2 rdf:type vcs: ActivitySchedule .
```

```
    […]
    vci:patientActivity2 vcs:goal ?new_goal .
    ?clinicalPathway vcs: scheduledActivity vci:patientActivity2 .
}.
```

Figure 35: Notation3 rule for adapting an activity

5. Conclusions and Next Steps

The deliverable presents and explains the final version of the vCare ontology, which comprises RDF schemata for vCare profiles (namely user, disease pattern, activity, treatment plan, health progress, device and coaching application), for decision-making applications (namely the state profile), as well as instances related to all four disease cases addressed in the vCare project. The deliverable illustrates how RDF can be successfully used to model appropriate classes for the vCare profiles and how to constrain the entered and generated instances using SHACL.

The final vCare ontology integrates the knowledge of prominent and established ontologies, such as RadLex, Human Disease Ontology, universAAL Health Measurement, Stroke Ontology, Logical Observation Identifiers Names and Codes, Gene Ontology, Medical Subject Headings, Semantic Sensor Ontology (and SOSA - Sensor, Observation, Sample, and Actuator), M3 (i.e. Machine-to-Machine Measurement), Product ontology, Friend-of-a-Friend and vCard.

Finally, the deliverable also presented the mechanism used to infer further knowledge from modelled facts by using Notation3 combined with expert rules modelled by medical experts. During the Tech lab phase there have been various consultations with the medical partners to define further expert rules that serve as a basis for the adaptation of the clinical pathway. Besides another set of rules that were added to the database, we were also able to develop a process to define further rules. This is particularly useful for the upcoming Living lab phase, as the Tech lab phase focussed on defining and testing all technical interfaces among partners. In the Living lab phase, we specifically target real scenarios with real patient data in more complex contexts. Building on the initial version in **D4.2**, the ontology schema is extended with required properties (i.e. relationships of activities) as well as links to technical partners components, such that all required information is modelled in the knowledge layer. The conversion mechanisms from raw data (e.g. sensor values or semi-structured information about patients) to semantic data as structured in the vCare ontology was tested during the TechLab phase and has proven that relevant data can be directly fed into the vCare ontology.

As ongoing work and with high focus during the Living lab phase, we continue to model further medical expert rules which go beyond D1.3 in order to improve the rule-based agent for recommending activities of clinical pathways. This will lead to risk-aware pathway adaptation, where only medical expertise is used. The enriched set of rules will also be the baseline for Machine Learning (ML)-based agents, especially using Reinforcement Learning (RL), where they are used to warm-start the resulting agents and to constrain their adaptation recommendations to safe options. The results of this work will be presented in **D4.4**.