Semantic Web 1 (2016) 1–5
IOS Press

REHABROBO-QUERY: Answering Natural Language Queries about Rehabilitation Robotics Ontology on the Cloud

Editor(s): Stamatia Dasiopoulou, Pompeu Fabra University, Spain; Georgios Meditskos, Centre for Research and Technology Hellas, Greece; Leo Wanner, ICREA and Pompeu Fabra University, Spain; Stefanos Vrochidis, Centre for Research and Technology Hellas, Greece; Philipp Cimiano, Bielefeld University, Germany
Solicited review(s): Antonis Bikakis, University College London, London, UK; Nick Bassiliades, Aristotle University of Thessaloniki, Thessaloniki, Greece; Eleni Kamateri, Center for Research and Technology Hellas (CERTH), Thessaloniki, Greece; Four anonymous reviewers

Zeynep Dogmus and Esra Erdem and Volkan Patoglu
Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, Turkey
E-mail: {zeynepdogmus,esraerdem,vpatoglu}@sabanciuniv.edu

Abstract. We introduce a novel method to answer natural language queries about rehabilitation robotics, over the formal ontology REHABROBO-ONTO. For that, (i) we design and develop a novel controlled natural language for rehabilitation robotics, called REHABROBO-CNL; (ii) we introduce translations of queries in REHABROBO-CNL into SPARQL queries, utilizing a novel concept of query description trees; (iii) we use an automated reasoner to find answers to SPARQL queries. To facilitate the use of our method by experts, we develop an intelligent, interactive query answering system, called REHABROBO-QUERY, using Semantic Web technologies, and make it available on the cloud via Amazon web services. REHABROBO-QUERY guides the users to express their queries in natural language and displays the answers to queries in a readable format, possibly with links to detailed information. Easy access to information on REHABROBO-ONTO through complex queries in natural language may help engineers inspire new rehabilitation robot designs, while also guiding practitioners to make more informed decisions on technology based rehabilitation.

Keywords: Ontology systems, query answering, rehabilitation robotics, intelligent user-interfaces, controlled natural languages

1. Introduction

Neurological injuries, such as stroke, are the leading cause of permanent disability in developed countries [1]. In particular, among 17 million people that suffer from stroke, about one third are left permanently disabled each year. These disabilities place a high burden on individual welfare of patients, while negatively impacting national economies [2]. Despite the recent medical developments, the number of stroke incidents continues to increase due to the ageing population in many developed countries.

Physical rehabilitation therapy is indispensable for treating neurological disabilities. Therapies have been shown to be more effective when they are task specific [5], repetitive [7], intense [29], long term [44], and allow for active involvement of patients [35]. However, repetitive and high intensity therapies place high physical burden on the therapist, significantly increasing the cost of such treatments.

With recent advancements in electro-mechanical systems, robot-assisted rehabilitation devices have become ubiquitous, since these devices can bear the physical burden of rehabilitation exercises, while therapists are employed as decision makers. In particular, the use of robotic devices in repetitive and physically involved rehabilitation eliminates the physical burden of movement therapy for the therapists, enables

safe and versatile training with increased intensity, and increases the reliability, accuracy, and effectiveness of traditional physical rehabilitation therapies. Robot-assisted rehabilitation devices can be used to support patients with all levels of impairment, can quantitatively measure patient progress, allow for easy tuning of duration and intensity of therapies, and enable realization of customized, interactive, innovative treatment protocols. Clinical trials with robot-assisted rehabilitation indicate that this form of therapy is effective for motor recovery and possesses high potential for improving functional independence of patients [30,37,40,41,47].

As more and more rehabilitation robots are deployed, the information about them also increases. However, this information is not represented as knowledge in structured forms, and usually appears as text in relevant publications. Consequently, accessing the requested knowledge and thus automatically reasoning about it has become a challenge. For instance, accessing the flexion/extension range of motion (RoM) of ASSISTON-SE [49], and determining the rehabilitation robots that target shoulder movements and also have at least 210° RoM for the flexion/extension movements of the shoulder are challenging tasks that require one to go through and study unstructured text from several different resources.

Furthermore, given the interdisciplinary nature of rehabilitation robotics, in many cases, the requested knowledge requires integration of further knowledge from related disciplines, such as physical medicine. For instance, determination of rehabilitation robots that can treat patients with rotator cuff lesions, requires one to know that rotator cuffs are muscle units employed to move the shoulder, and that, for patients with rotator cuff lesions, abduction and flexion movements of the shoulder should not have more than 90° RoM. Relevant rehabilitation robots can only be found after one acquires this crucial information.

Additionally, given the growing number of research groups contributing to the field, different approaches display large variability and the field lacks a standardized terminology. Several efforts have been initiated for standardizing terminology, as well as assessment measures, for rehabilitation robots, e.g., by the European Network on Robotics for Neurorehabilitation.[1] The development of such a standardization is likely a critical step for the field, helping robotic rehabilitation technology become widely understood and accepted as a useful tool.

Motivated by these challenges and efforts, we have earlier designed and developed the first formal rehabilitation robotics ontology, called REHABROBO-ONTO, in OWL (Web Ontology Language) [3,25], and made it available on the cloud via Amazon Web Services [13,14,16] (in particular, Amazon EC2 – Elastic Compute Cloud)[2] so that every rehabilitation robotics researcher can easily use it in order to publish/represent information about his/her robot, and modify this information. To facilitate such modifications of the rehabilitation robotics ontology REHABROBO-ONTO, we have also developed a Web-based software (called REHABROBO-QUERY)[3] with an intelligent user-interface. In this way, experts do not need to know the underlying logic-based representation languages of ontologies, like OWL, or Semantic Web technologies, for information entry and modification. REHABROBO-QUERY utilizes Amazon Web Services (Amazon EC2) for cloud computing. For further information about the design, development and maintenance of REHABROBO-ONTO, and how REHABROBO-QUERY facilitates modification of REHABROBO-ONTO, we refer the reader to our earlier article [16].

This article addresses question answering in the domain of rehabilitation robotics over REHABROBO-ONTO. Note that a structured representation of information about rehabilitation robotics such as the ontology REHABROBO-ONTO allows rehabilitation robotics researchers to learn various properties of the existing robots and access the related publications to further improve the state-of-the-art. Physical medicine experts also can find information about rehabilitation robots and related publications to better identify the appropriate robot for a particular therapy or patient population. Such requested information can be obtained from REHABROBO-ONTO by expressing the requested information as a formal query in a query language, such as SPARQL [42], and then by computing answers to these queries by using a state-of-the-art automated reasoner, such as PELLET [43]. However, expressing the requested information as a formal query by means of formulas is challenging for many users, including robot designers and physical medicine ex-

[1] http://www.cost.eu/COST_Actions/bmbs/TD1006

[2] http://aws.amazon.com/ec2/

[3] http://hmi.sabanciuniv.edu/?page_id=781

perts. For that reason, we particularly focus on the process of query answering over REHABROBO-ONTO, and making it easier for the users to express their queries in a natural language and to obtain answers to their queries automatically.

Towards these goals, the main contributions of this article can be summarized as follows. For expressing queries about rehabilitation robotics, we have designed and developed a novel controlled natural language for rehabilitation robots, called REHABROBO-CNL. For automatically computing answers to natural language queries, we have introduced an algorithm that transforms natural language queries in REHABROBO-CNL into formal queries in SPARQL. This transformation utilizes an intermediate representation: a novel tree structure, called a Query Description Tree (QDT). Once the natural language query is transformed into a formal query, PELLET is used to find relevant answers to the query. The software system REHABROBO-QUERY has been extended with an interactive, intelligent user interface to guide the users to enter natural language queries in REHABROBO-CNL about the existing robots, and to present the answers in an understandable form, so that the users do not have to know about the logical formalism of the ontology or the formalism to represent queries, or the use of the technologies for computing answers to their questions about rehabilitation robots.

The rest of the article is organized as follows. First, we present a brief review of REHABROBO-ONTO (Section 2). We present the controlled natural language REHABROBO-CNL for expressing queries about rehabilitation robotics (Section 3), and discuss the extension of REHABROBO-QUERY to support querying in REHABROBO-CNL via an interactive, intelligent user-interface (Section 4). We present our transformation of a REHABROBO-CNL query into a SPARQL query (Section 5) whose answer can be computed using PELLET (Section 6). We evaluate the underlying methods empirically over a variety of queries, and the usefulness of REHABROBO-QUERY by a survey analysis with participants of different backgrounds (Section 7). After we summarize the related work about software systems that support natural language queries over ontologies (Section 8), we conclude with a summary of our contributions (Section 9).

This article significantly extends our earlier paper [15], presented at the International Conference on Knowledge Engineering and Ontology Development (KEOD 2014), (i) by providing more details about the query answering system integrated in REHABROBO-QUERY, the types of queries supported by REHABROBO-QUERY, and the complete description of the query language REHABROBO-CNL, (ii) by introducing a new algorithm for transformation of a REHABROBO-CNL query into a SPARQL query, and (iii) by providing evaluations of the underlying methods and the query system REHABROBO-QUERY.

2. A Brief Review of REHABROBO-ONTO

REHABROBO-ONTO is the first formal rehabilitation robotics ontology that represents knowledge about rehabilitation robotics in a structured form, and allows query answering about this knowledge. It has been developed in line with the efforts of the European Network on Robotics for Neurorehabilitation,[4] for standardizing terminology as well as assessment measures for rehabilitation robots.

REHABROBO-ONTO has been designed by considering suggestions of various rehabilitation robotics researchers and physical medicine experts, by first identifying the purpose, and then the basic concepts, their thematic classes and their relationships. The main classes and their relationships are illustrated in Figure 1.

REHABROBO-ONTO has five main concepts (or thematic classes):

- `RehabRobots` (representing rehabilitation robots and their properties),
- `JointMovements` (representing targeted joint movements and their properties),
- `Owners` (representing robot designers who add and modify information in the ontology about their own robots),
- `References` (representing publications related to rehabilitation robots),
- `Assessments` (representing assessment measures for rehabilitation robots).

The main classes `RehabRobots`, `JointMovements` and `Assessments` have further subclasses. For instance, `JointMovements` has two main subclasses, for lower extremity joint movements and upper extremity joint movements; Figure 3 of [16] illustrates a hierarchy of the lower extremity joint movements, while Figure 2 shows a hierarchy of the upper extremity joint movements. A hierarchy of various types

[4] `http://www.rehabilitationrobotics.eu/`

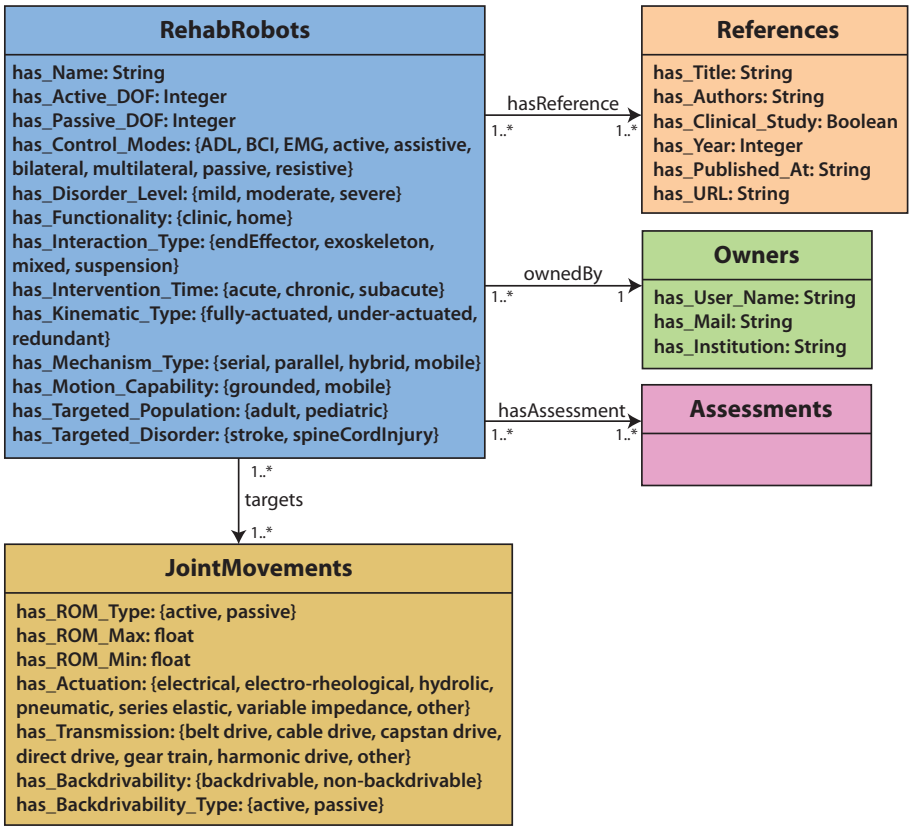


Fig. 1. REHABROBO-ONTO with main classes [16, Figure 1].

of assessment measures (e.g., movement quality assessments, effort assessments, psychomotoric assessments, muscle strength assessments, kinematic assessments) is shown in Figure 4 of [16]. Currently there are 147 classes represented in REHABROBO-ONTO.

The main concepts are related to each other by the following relations:

- a rehabilitation robot `targets` joint movements,
- a rehabilitation robot is `ownedBy` a robot designer,
- a rehabilitation robot `hasReferences` to some publications,
- a rehabilitation robot `hasAssessment` with respect to some evaluation measure.

REHABROBO-ONTO is an OWL 2DL ontology that has been developed using the ontology editor PROTÉGÉ [24]. It is open-source and available on the cloud via Amazon Web Services. Its maintenance (i.e., adding, deleting, modifying information about rehabilitation robots) can be done as part of the ontology system REHABROBO-QUERY.

REHABROBO-ONTO has been designed to be as simple as possible for the purpose of easier maintainability and the possibility of extensions with further domain-specific properties when needed, yet as functional as needed by the experts. For instance, since the main requested information is essentially about the robot, the basic bibliography information and the domain-specific information about clinical studies are found sufficient by the experts. Therefore, `References` has been described as one simple class containing this information only, instead of utilizing BIBO [11] or FaBio [39] that can represent further details about publications in many classes. Similarly, for `Owners`, it is sufficient to introduce one class with few data properties instead of utilizing FOAF.[5]

For further information about the design and development of REHABROBO-ONTO, and how REHABROBO-QUERY facilitates maintenance of REHABROBO-ONTO

[5] `http://www.foaf-project.org/`.

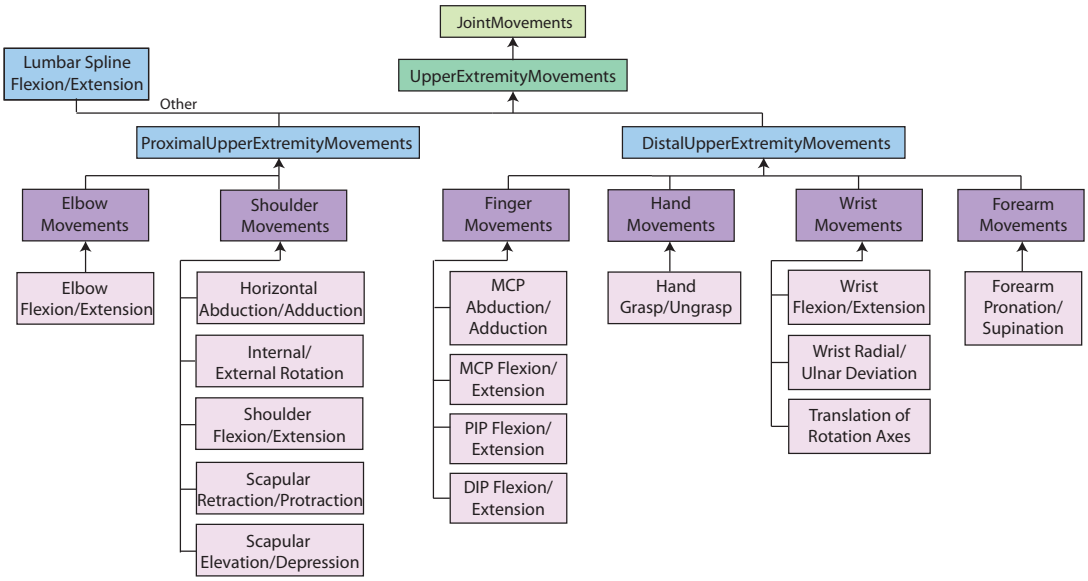


Fig. 2. A hierarchy of upper extremity joint movements targeted by rehabilitation robots.

on the cloud, we refer the reader to our earlier article [16].

3. REHABROBO-CNL: A Controlled Natural Language for Queries about Rehabilitation Robotics

Queries about REHABROBO-ONTO can be posed by the user in a natural language. However, natural languages may have ambiguities in their vocabularies and grammars. To overcome ambiguities of a natural language, a subset of it with a restricted vocabulary and grammar can be considered for specific domains; such languages are called Controlled Natural Languages (CNLs) [28]. Essentially, with its restricted vocabulary and grammar, a CNL is a formal language. Therefore, it is easier to convert a CNL (compared to a more general natural language) to a logic-based formalism. In that sense, a CNL facilitates the use of automated reasoners to find answers to queries expressed in a CNL. Due to these reasons, first we have identified the types of queries about rehabilitation robots, then we have designed and developed a new CNL, called REHABROBO-CNL, to express these queries in natural language.

3.1. *Types of queries supported by* REHABROBO-CNL

The variety of queries is important since the users may include not only rehabilitation robot designers and developers, but also physical therapists and medical doctors. Also, complex queries that not only require intelligent extraction of different types of knowledge, but their integration via conjunctions, disjunctions, negations, nested relative clauses, aggregates, and quantifications are inevitable in rehabilitation robotics, since robots, targeted movements, evaluation metrics, etc. have many features that are of interest to users. We have designed REHABROBO-CNL considering these aspects.

We have identified different types of complex queries useful from the perspectives of rehabilitation robotics and physical medicine, by gathering sample queries from a large number of experts from different disciplines (i.e., roboticists, engineers, physical therapists, physical medicine doctors, neuroscientists) whom we have been closely collaborating with and whom we have met at various meetings and workshops. We have further populated such queries (e.g., Q6 and Q19 of Table 8) considering their grammatical structures, for the purpose of testing and evaluating the CNL. Some examples of these queries are presented in the following, with respect to their topics.

Queries about mechanical properties of rehabilitation robots These are queries to extract information about

the rehabilitation robots whose information is available in REHABROBO-ONTO. Here are some examples:

- Q1 What are the robots that target shoulder movements and that have at least 210° RoM for the flexion/extension movements of the shoulder?
- Q2 What are the robots that target wrist movements and that have at least 2 active degrees of freedom?
- Q3 What are the shoulder robots that target flexion/extension movements and that do not target elevation/depression movements?
- Q4 What are the robots with active degree of freedom $\geq$ '3' and that target all pelvic girdle movements with backdrivability type 'passive'?
- Q5 What are the ankle robots that target movements with electrical actuation and with cable drive transmission?

According to REHABROBO-ONTO, for instance, some part of the answer to Q1 is as follows:

ASSISTON-SE

Query Q1 is useful for an engineer to determine rehabilitation robots that can cover the whole range of motion of shoulder flexion/extension movements. Existing designs may not only guide new solutions for the same joint, but similar solutions may be applied to different joints of the body. For instance, ASSISTON-SE [49] and ASSISTON-GAIT [36] rehabilitation robots rely on a similar mechanism design to target scapula-shoulder and pelvis-hip joint complexes. On the other hand, Q1 can also be used by a therapist to identify robots that can be used to target dysfunction of the scapulohumeral rhythm, as the ability to ergonomically cover large flexion/extension range of motions is critical for such treatment.

Similarly, an engineer may benefit from Q3 to determine robot designs that can align robot and human shoulder rotation axes and use these examples to design other robots that have similar alignment properties. Q3 may be used by a therapist to identify shoulder robots that cannot be used to treat shoulder pain due to shoulder subluxation, as elevation/depression movements are critical for treatment of this medical condition.

Queries about joint movements targeted by rehabilitation robots These are queries to extract information about joint movements targeted by rehabilitation robots in REHABROBO-ONTO. Here are some examples:

- Q6 What are the movements that are targeted by some robots with (some intervention time or with all targeted disorders)?
- Q7 What are the movements that are targeted by the robot 'AssistOn-Finger' and with minimum range of motion $\geq$ '20'?
- Q8 What are the movements that are not targeted by any robots with kinematic type = 'redundant' and with mechanism type $\neq$ 'parallel'?

According to REHABROBO-ONTO, for instance, some part of the answer to Q7 is as follows:

Index Finger DIPFlexion/Extension
Ring Finger PIPFlexion/Extension
Pinky Finger DIPFlexion/Extension
Thumb Finger PIPFlexion/Extension
Middle Finger PIPFlexion/Extension

Query Q7 may be used by a therapist while evaluating whether ASSISTON-FINGER [19], which is a robot designed for tendon injuries, is also appropriate for finger range of motion exercises for stroke patients.

Query Q8 is useful for an engineer to determine joint movements that are targeted with redundant but not parallel mechanisms. Such a query may help determine similar characteristics of several joint movements and guide similar mechanism designs to be adapted for these joints.

Queries about the evaluation metrics for rehabilitation robots These are queries to extract information about metrics used to evaluate rehabilitation robots. Here are some examples:

- Q9 What are the effort metrics that are evaluated by some robots with active degree of freedom $\geq$ 2?
- Q10 What are the movement quality metrics that are evaluated by all robots with motion capability = 'grounded'?
- Q11 What are the kinematic aspect metrics that are evaluated by some robots that target all elbow movements?
- Q12 What are the muscle strength metrics that are evaluated by robots that target all wrist movements with transmission = 'direct drive'?
- Q13 What are the psychomotoric aspect metrics that are evaluated by all robots with kinematic type = 'redundant' and that target all ankle movements with minimum range of motion $\geq$ '30'?

According to REHABROBO-ONTO, for instance, some part of the answer to Q9 is as follows:

Time To Initiate Movement
Amount Of Compensation
Biomechanical Work Energy Power

An engineer may use Q9 to determine the types of sensors to be integrated in a new multi degree of freedom rehabilitation robot, while a clinical researcher may utilize the same query to identify robots that can provide relevant effort metrics to study abnormal synergy patterns after stroke.

Similarly, Q11 is useful for an engineer who designs an elbow robot to determine the relevant kinematic aspect metrics that can be incorporated into the control/evaluation algorithms of a new design. The same query can be used by therapists to identify which kinematic aspect metrics can be measured by current robotic rehabilitation devices.

Other queries These are queries to extract information about publications and owners/institutes/laboratories of the robots. Here are some examples:

- Q14 What are the publications with clinical study and that do not reference any robots with active degree of freedom ≥ 1?
- Q15 What are the publications without clinical study or that reference some robots that do not evaluate any movement quality metrics?
- Q16 What are the publications with place of publication 'ICORR' and that reference some robots that are owned/maintained by some users with institution 'Sabanci University'?
- Q17 What are the users that own/maintain some robots that target all ankle movements?

Queries Q16 and Q17 may be used by both engineers and clinical researchers to find publications of a research group or research groups that focus on certain joint movements.

3.2. *Grammar of* REHABROBO-CNL

Once we have identified different types of complex queries about rehabilitation robotics, we have designed the controlled natural language REHABROBO-CNL for expressing them in an unambiguous way and utilizing domain-specific vocabulary. The grammar of REHABROBO-CNL, with relevant vocabulary, is presented in Tables 1–7.

Table 1 gives an overall summary of REHABROBO-CNL, providing information about what types of queries are allowed in REHABROBO-CNL. To eliminate the ambiguities in nesting of conjunctions and disjunctions, REHABROBO-CNL provides two ways of constructing a query: A query in REHABROBO-CNL should either be in Conjunctive Normal Form (CNF), or in Disjunctive Normal Form (DNF). In other words, REHABROBO-CNL supports conjunctions of simple disjunctions, and disjunctions of simple conjunctions. No further nesting of conjunctions (resp., disjunctions) in a simple disjunction (resp., conjunction) is allowed. A query can contain any number of conjunctions and disjunctions provided that they confirm with the rules above. An example of a query in CNF is as follows.

> What are the robots with mechanism type='hybrid' and (with motion capability ='grounded' or with functionality='clinic')?

The following query is in DNF:

> What are the robots with no targeted disorder or (with active degree of freedom > 1 and with control modes='active')?

Having an overall understanding of the types of queries supported by REHABROBO-CNL, let us provide some more details to relate it to rehabilitation robotics. The functions presented in italic font in Table 1 refine these queries by embedding relevant information from REHABROBO-ONTO. These ontology functions are described in Table 2.

The information represented with the ontology functions are coupled by their relevance. For instance, only the verb "reference" can appear after the type Publications. By such a matching of types with verbs, it is possible to prevent semantically wrong queries like "What are the publications that target some shoulder movements?". All of the matches between types and verbs in expressions of the form "$Type()$ that $Verb()$" are shown in Table 3. Note that these matchings essentially come from the structure of REHABROBO-ONTO, e.g., concept names and relation names.

Similarly, it is necessary to match verbs with types in expressions of the form "$Verb()$ (some | all | any* | the) $Type()$". Table 4 lists the available types that can occur after a verb in the query (e.g., in a RELATIVE-CLAUSE), relative to the ontology. If a quantifier such as "some" is used in a relative clause, then the types which have some subclasses are considered in queries. Here is an example query:

> What are the robots that evaluate some wrist movements?

Since wrist movements class have subclasses (e.g., wrist flexion/extension, wrist radial deviation/ulnar de-

Table 1
The Grammar of REHABROBO-CNL

QUERY →	WHATQUERY QUESTIONMARK
WHATQUERY →	What are the *Type*() GENERALRELATION
GENERALRELATION →	SIMPLERELATION NESTEDRELATION*
SIMPLERELATION →	(that RELATIVECLAUSE)+
SIMPLERELATION →	that INSTANCERELATION
SIMPLERELATION →	WITHRELATION
NESTEDRELATION →	(and LP SIMPLEDISJUNCTION RP)*
NESTEDRELATION →	(or LP SIMPLECONJUNCTION RP)*
SIMPLEDISJUNCTION →	(SIMPLERELATION or)* SIMPLERELATION
SIMPLECONJUNCTION →	(SIMPLERELATION and)* SIMPLERELATION
RELATIVECLAUSE →	*Verb*() (some | all | the) *Type*()
RELATIVECLAUSE →	NEG *Verb*() any *Type*()
INSTANCERELATION →	NEG? *Verb*() the *Type*() *Instance*()
WITHRELATION →	with *Noun*() EQCHECK *Value*()+
WITHRELATION →	with QUANTIFIER *Noun*()
WITHRELATION →	(with | without) *Noun*()
EQCHECK →	$= \mid != \mid \leq \mid \geq$
QUANTIFIER →	some | all | none
NEG →	*Neg*()
LP →	(
RP →	)
QUESTIONMARK →	?

Table 2
The Ontology Functions

Type()	Returns the types that correspond to concept names. They are: Robots, movements, users, publications and metrics.
Instance()	Returns robot names for robots and user names for users.
Verb()	Returns the verbs that correspond to object properties between concepts. Returns both active and passive forms of these verbs. Active forms of these verbs are: Target, evaluate, reference, own.
Noun()	Returns the nouns that correspond to data properties. ex. targeted disorder, active degree of freedom.
Value()	Returns the suitable values according to a given noun. Corresponds to the pre-defined ranges of data properties.
Neg()	Returns a suitable negation phrase. These phases are: do not, are not.

viation) in REHABROBO-ONTO, this query will retrieve all robots that target at least one of these subclasses. If "the" keyword is used after the verb in a query, then the leaf classes are considered to select one specific type. Here is an example query:

> What are the robots that target the wrist radial deviation/ulnar deviation?

Wrist radial deviation/ulnar deviation is a leaf class. It is also a subclass of wrist movements. This query will retrieve the robots that target this specific wrist movement. If there is a robot that targets some other wrist movements but wrist radial deviation/ulnar deviation, then this robot will not be included in the answer to this query.

In REHABROBO-CNL, the instances of the concepts are represented by one of their distinctive properties. For robots, this distinctive property is its name; for users, it is the user name. To illustrate, when the users want to query about ASSISTON-SE, they specify the instance using the name of the robot. For movements and metrics, there is no such distinctive property because the concept names are sufficient to specify a movement or metric. In fact, using RELATIVECLAUSE is sufficient to query about them. To query about robots or users, however, INSTANCERELATION is used to extract the instances. Table 5 demonstrates the relevant

Table 3
Verbs that can occur after the nouns

Type()	that	*Verb*()
robots	→	target
robots	→	are owned/maintained
robots	→	evaluate
movements	→	are targeted by
users	→	own/maintain
publications	→	reference
effort metrics	→	are evaluated by
kinematic aspect metrics	→	are evaluated by
movement quality metrics	→	are evaluated by
muscle strength metrics	→	are evaluated by
psychomotoric aspect metrics	→	are evaluated by

Table 4
Types that can occur after the verbs

Verb()	(some | all | any* | the)	*Type*()
target	(some | all | any)	all movements except leaf classes
target	the	leaf classes of movements
evaluate	(some | all | any)	all metrics except leaf classes
evaluate	the	leaf classes of metrics
are targeted by	(some | all | any)	robots
are evaluated by	(some | all | any)	robots
reference	(some | all | any)	robots
own	(some | all | any)	robots
are owned by	(some | all | any)	users

* *any* is used after negative verbs.

Table 5
Instances that can occur after the types

Type()		*Instance*()
robot	→	name of the robot
user	→	username of the user

properties of the instances that appear when a type is selected.

In addition to types and verbs, types are matched with the relevant nouns, as shown in Table 6. For instance, control modes are matched with robots whereas actuation is matched with movements. Note that these matchings are due to properties of concepts in REHABROBO-ONTO.

Further, the values for the nouns are extracted from REHABROBO-ONTO to allow suitable entries from the users. These values are listed in Table 7. The values can be considered as ranges of the nouns, that the user can choose from.

4. User-Interface of REHABROBO-QUERY for Query Answering: Intelligent and Interactive

The user interface of REHABROBO-QUERY can guide the users to add and modify information in REHABROBO-ONTO. We have extended it further so that it can guide the users to ask questions in REHABROBO-CNL, and present answers to these queries in an understandable form.

While the users construct questions, REHABROBO-QUERY's user interface can prevent nonsensical queries. For instance, shoulder elevation/depression is not a wrist movement. The user interface takes into account such information, and guides the users intelligently to construct their queries so that queries like

> What are the wrist robots that target shoulder elevation/depression?

are not possible. In that sense, the user interface is intelligent.

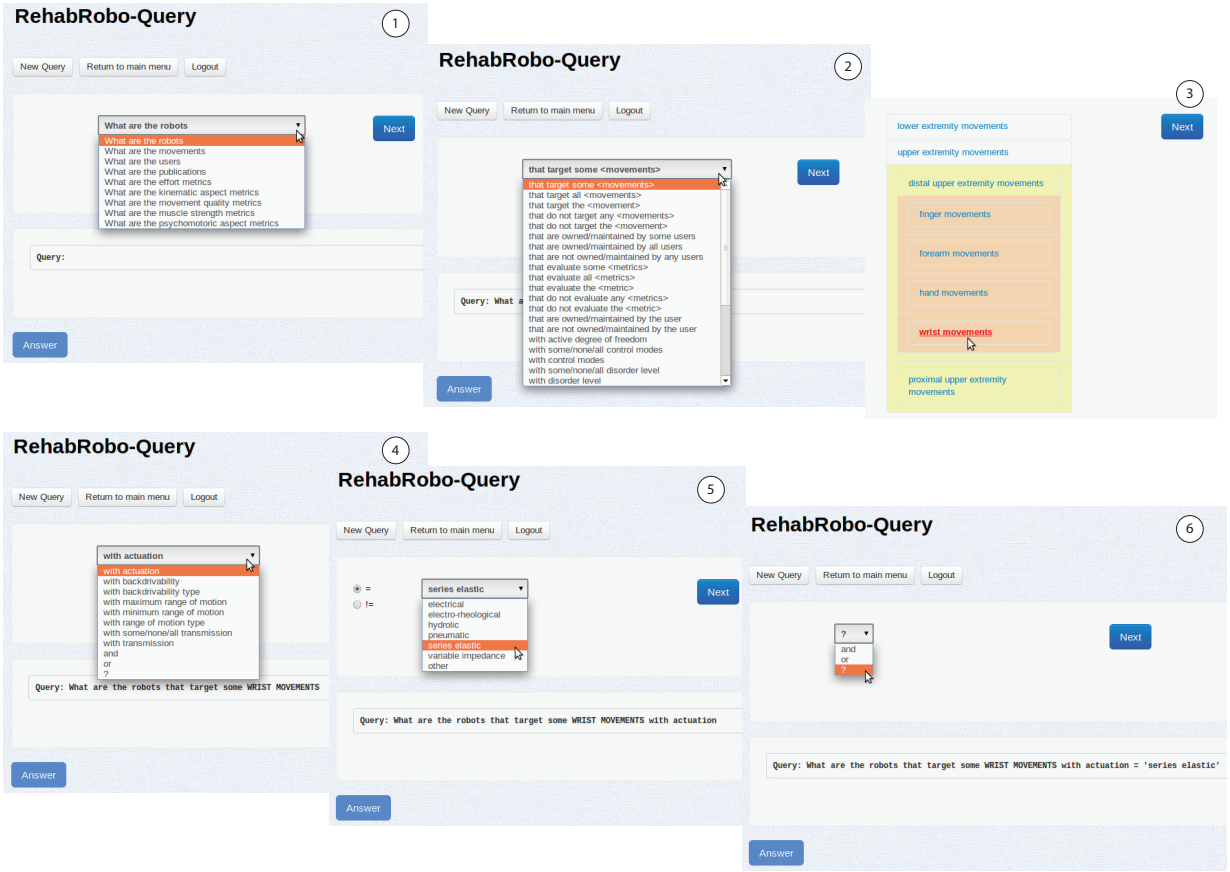


Fig. 3. Construction of a query with the guidance of the intelligent interactive user interface of REHABROBO-QUERY.

REHABROBO-QUERY's user interface is also interactive: it shows the possible choices (for instance, the sort of movements targeted by a shoulder robot) and allows auto-completion (for instance, to obtain the full name of a robotics device). To be able to identify the choices (e.g., parameters and their values) relevant to the query, REHABROBO-QUERY's user interface utilizes automated reasoning methods online. Indeed, as the user enters a query, in the background the user interface considers the part of the query constructed so far and asks REHABROBO-QUERY's ontological reasoning module to compute all possible values of parameters. Once the reasoning module returns these choices, the user interface presents them, e.g., within a pull-down menu. Figure 3 shows the construction of the following query with the guidance of the user interface:

> What are the robots that target some wrist movements with actuation='series elastic'?

Fig. 4. An answer for the query constructed in Figure 3, presented to the user.

Furthermore, REHABROBO-QUERY provides auto-completion to help users enter values for nouns that correspond to data properties of type string. If the user should choose a concept among a hierarchy, REHABROBO-QUERY displays an accordion view and enables the user to click on the desired option. In addition, REHABROBO-QUERY allows multiple selection of values for relational properties. For functional properties, the user is able to select multiple items for

Table 6
Nouns that can occur after the types

Type()	with	*Noun*()
robots	→	active degree of freedom
robots	→	control modes
robots	→	disorder level
robots	→	functionality
robots	→	interaction type
robots	→	intervention time
robots	→	kinematic type
robots	→	motion capability
robots	→	name
robots	→	passive degree of freedom
robots	→	targeted disorder
robots	→	targeted population
movements	→	actuation
movements	→	backdrivability
movements	→	backdrivability type
movements	→	maximum range of motion
movements	→	minimum range of motion
movements	→	range of motion type
movements	→	transmission
publications	→	authors
publications	→	clinical study
publications	→	place of publication
publications	→	title
publications	→	url
publications	→	year
users	→	institution
users	→	mail
users	→	username

inequality. The user can choose a number of options among "less than or equal", "more than or equal", "equal" and "not equal" while entering values for a data property of type integer or float.

Once the query is constructed, REHABROBO-QUERY asks the reasoning module to compute answers, and presents answers to queries concisely. It also includes links to detailed information in case the user wants further information. The answer to the query constructed in Figure 3 is shown to the user as in Figure 4.

Note that how the results of a query is displayed to user depends on the type of the query. For instance, if the query is about robots, then the user sees the names of the retrieved robots. If the query is about movements or metrics, then the user sees the concept names instead of the instance URIs which would make no sense to the user.

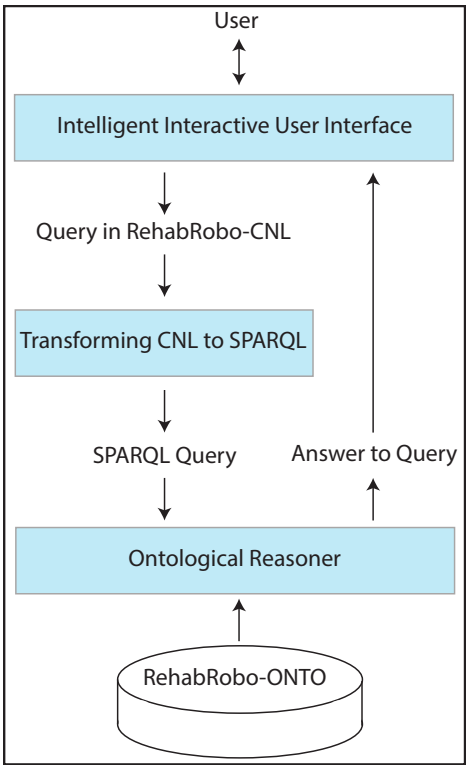


Fig. 5. An overview of query construction in REHABROBO-QUERY.

5. Transforming a Query in REHABROBO-CNL to a SPARQL Query

The process of answering a query in REHABROBO-CNL, illustrated in Figure 5, starts with a transformation of the query into a SPARQL query. We propose a transformation with the following steps.

1. We parse the query and form a query description tree.
2. We traverse the tree and form a SPARQL query.

The methodology of the first step of this transformation is domain-independent if the ontology functions are provided. Note that the ontology functions to extract relevant information from ontologies, so as to match the grammar and the vocabulary of REHABROBO-CNL, may need to be defined differently for other ontologies since the concepts/relations may be structured and named differently. The second step of the transformation, from query description trees to SPARQL queries, is domain-independent as well.

5.1. Query Description Trees (QDT)

We introduce a rooted, directed tree, called query description tree (QDT), to parse the REHABROBO-

Table 7
Values that can occur after the nouns

Noun()	*	*Value*()
active DoF**	→	any integer value entered by the user
actuation	→	electrical, electro-rheological, hydraulic, pneumatic, series elastic, variable impedance, other
authors	→	one of the authors that are added to the ontology up to now
backdrivability	→	backdrivable, non-backdrivable
backdrivability type	→	active, passive
control modes	→	ADL, BCI, EMG, active, assistive, bilateral, multilateral, passive, resistive
disorder level	→	mild, moderate, severe
functionality	→	clinic,home
institution	→	one of the institutions that are added to the ontology up to now
interaction type	→	exoskeleton, mixed, suspension, end effector
intervention time	→	acute, chronic, subacute
kinematic type	→	hybrid, parallel, serial
motion capability	→	grounded, mobile
name	→	one of the robot names that are added to the ontology up to now
passive DoF	→	any integer value entered by the user
place of publication	→	one of the places of publication that are added to the ontology up to now
maximum RoM***	→	any float value entered by the user
minimum RoM	→	any float value entered by the user
RoM type	→	active, passive
targeted disorder	→	stroke, spine cord injury
targeted population	→	adult, pediatric
title	→	one of the publication titles that are added to the ontology up to now
transmission	→	belt drive, cable drive, capstan drive, direct drive, gear train, harmonic drive, other
url	→	one of the urls that are added to the ontology up to now
username	→	one of the usernames that are added to the ontology up to now
year	→	any year (integer value) entered by the user

* (EQCHECK|QUANTIFIER) ** degree of freedom *** range of motion

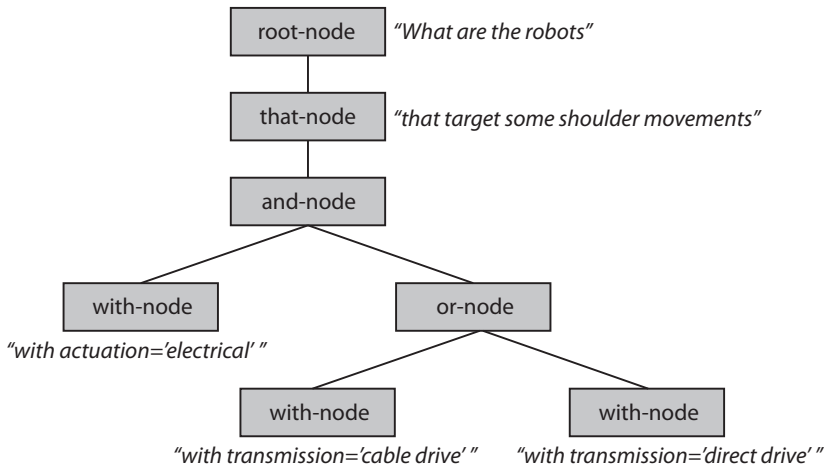


Fig. 6. Tree representation of the sample query.

CNL query entered by the user. In this tree, there are five types of nodes:

– root-node: Represents the type of the query.

- that-node: Represents a relative clause beginning with "that".
- with-node: Represents a relative clause beginning with "with".
- and-node: Represents a conjunction.
- or-node: Represents a disjunction.

Every root/that/with-node characterizes a phrase and a type/instance. An and/or-node cannot be a leaf. For each path from the root node to a leaf node, there do not exist three or more consecutive and/or-nodes. With-nodes are leaves only. That-node has one child only.

Every QDT is constructed online by a single pass over the input REHABROBO-CNL query guided by an interactive user-interface relative to the REHABROBO-CNL grammar. Therefore, the construction of a QDT from a REHABROBO-CNL query takes linear time in the size of the query (i.e., the number of nodes in QDT).

Consider, for instance, the QDT in Figure 6 for the query:

> What are the robots that target some shoulder movements with actuation='electrical' and (with transmission = 'cable drive' or with transmission='direct drive')?

This tree is constructed online while the user expresses the query by the help of the user interface. The root denotes the beginning of the query "What are the robots...". According to the root, the answer to the query will contain robot names only. Since the query is about robots, the type contained in the root is "robot".

The relative clause about these robots is the child of the root, and since this relative clause starts with "that", it is a that-node. The type contained in this node is "shoulder movement".

The query continues with a conjunction of relative clauses, including a simple disjunction. Clauses joined with a conjunction (resp., disjunction) are characterized by an and-node (resp., or-node) as their parent. Since these relative clauses start with "with", they are with-nodes. They include values of properties instead of types.

More precisely, every root-node, and-node, and or-node n is associated with

- *n.type*: the class it describes (e.g., `RehabRobots`),
- *n.children*: its children nodes.

Every that-node n is associated with

- *n.type*: the class it describes, i.e., the class of its parent node (e.g., `RehabRobots`),
- one of the following (according to REHABROBO-CNL):
  - *n.verb*, *n.quantifier* and *n.verb.type*: object property of this class (e.g., `targets`) in relation with `some/all/the` another class (e.g., `shoulderMovements`),
  - *n.verb*, *n.verb.type* and *n.verb.instance*: an object property of this class (e.g., `ownedBy`) in relation with an instance (e.g., $'$`OwnerX`$'$) of the relevant class (e.g., `Owners`),
- *n.child*: its child node,
- *n.active*: an identifier to specify that the verb describes an object property in active voice or passive voice,
- *n.negation*: an identifier to specify whether the condition specified by the that-node is negative.

Every with-node n is associated with

- *n.type*: the class it describes (e.g., `RehabRobots`),
- one of the following (according to REHABROBO-CNL):
  - *n.noun*, *n.operator* and *n.noun.value*: a data property of *n.type* (e.g., `hasFunctionality`) specified by a value (e.g., $'$`clinic`$'$) using a relevant operator ('=', '<=', '>=', '!='),
  - *n.noun*, *n.noun.quantifier*, and *n.noun.range*: a data property of *n.type* specified by some/all/none of a set of values.

5.2. *From QDT to a* SPARQL *query*

The tree representing the query, in fact, characterizes a concept. Algorithm 1 generates this concept in SPARQL by a depth-first traversal of a QDT. As the algorithm traverses a QDT, according to the types of nodes, it generates parts of the concept in SPARQL. Since the transformation is done by a depth-first traversal of a QDT and thus every node is visited once, it takes linear time in the size of the QDT (i.e, number of nodes).

Let us explain Algorithm 1 by some examples.

Example 1. Suppose that the input is the tree in Figure 6, the concept described by the tree is denoted by a variable `?x`, and the prefix of REHABROBO-ONTO is `pn`.

The algorithm starts with the `root-node` of the tree. Since the associated class is "robot", our de-

Algorithm 1: *transform*

Input : A QDT characterized by its root node n representing a concept, a prefix name pn of the ontology, a variable x
Output: A SPARQL construct Q that represents the concept in n

if *n is a root-node* **then**
  $Q \leftarrow transformClass(n.type, pn, x)$;
  $Q \leftarrow$ $intersect(Q, transform(n.child, pn, x))$;
else if *n is a that-node* **then**
  $Q \leftarrow transformThatNode(n, pn, x)$;
else if *n is a with-node* **then**
  $Q \leftarrow transformWithNode(n, pn, x)$;
else if *n is an and-node* **then**
  $Q \leftarrow null$;
  foreach *child node* $c \in n.children$ **do**
    $Q \leftarrow intersect(Q, transform(c, pn, x))$;
else if *n is an or-node* **then**
  $tQ \leftarrow null$;
  foreach *child node* $c \in n.children$ **do**
    $Q \leftarrow union(Q, transform(c, pn, x))$;
return Q

Algorithm 2: *transformClass*

Input : A class C of the ontology, the prefix name pn of the ontology, a variable x to denote C
Output: A SPARQL construct that represents C

return `?x rdf:type pn:C`

scription in SPARQL starts by specifying the class `RehabRobots` for variable ?x by Algorithm 2:

```
?x rdf:type rr:RehabRobots
```

Then the algorithm calls *transform* for the child of the root node, which is a `that-node`. It later calls *transformThatNode* (Algorithm 3) and passes `that-node` as an input, as well as the prefix name `rr` and the variable ?x. Since `that-node` describes ?x "that targets some shoulder movements", Algorithm 3 first calls *some* (Algorithm 4) and generates the following SPARQL expression:

```
?x rr:targets ?y .
```

Algorithm 3: *transformThatNode*

Input : A that-node n, the prefix name pn of the ontology, a variable x
Output: A SPARQL construct Q that represents the concept in n

if $n.quantifier = THE$ **then**
  if *n.verb.instance is not null* **then**
    // The relative clause is about an instance
    // e.g., [robots] that are ownedBy
    // the owner 'Dr.X'
    $Q \leftarrow objProp(x, pn, n.verb, n.verb.instance, n.active)$;
  else
    // The relative clause is about a leaf class
    // e.g., [robots] that target the knee flexion
    // extension movements
    $y \leftarrow$ generate a new variable;
    $Q_1 \leftarrow$ $objProp(x, pn, n.verb, y, n.active)$;
    $Q_2 \leftarrow$ $transformClass(y, pn, n.verb.type)$;
    $Q_3 \leftarrow transform(n.child, pn, y)$;
    $Q \leftarrow intersect(Q_1, Q_2, Q_3)$;
else if $n.quantifier = ALL$ **then**
  // e.g., [robots] that target all wrist
  // movements [with ...]
  $Q \leftarrow$ $forall(n.verb, n.verb.type, pn, x, n.child)$;
else
  // e.g., [robots] that target some wrist
  // movements [with ...]
  $Q \leftarrow$ $some(n.verb, n.verb.type, pn, x, n.child)$;
if $n.negation = True$ **then**
  $Q \leftarrow complement(Q)$;
return Q

```
?y rdf:type rr:ShoulderMovements
```

This expression describes the object property "targets" in the first line above (Algorithm 5), relating rehabilitation robots with some shoulder movements in the second line above (Algorithm 2). These two lines are combined by Algorithm 6.

The shoulder movements are further refined with a conjunction of specific data properties, which is de-

Algorithm 4: *some*

Input : An object property R (i.e., *n.verb*) associated with the class of a that/with-node (i.e., *n.type*), a class C (i.e., *n.verb.type*), a prefix name *pn* of the ontology, a variable x, a child node m (i.e., *n.child* – possibly null)
Output: A SPARQL construct that represents the existential restriction of C relative to R

// e.g., [robots that target] some wrist movements
// *n.type* is "robots", R is "targets",
// C is "wrist movements"

$y \leftarrow$ generate a new variable;
$Q_1 \leftarrow transformClass(y, pn, C)$;
$Q_2 \leftarrow objProp(x, pn, R, y, n.active)$;
$Q \leftarrow intersect(Q_1, Q_2)$;
if *m is not null* **then**
  $Q \leftarrow intersect(Q, transform(c, pn, y))$;
return Q

// the output for the example above:
// `?y rdf:type pn:C .`
// `?x pn:R ?y`

Algorithm 5: *objProp*

Input : An object property R, the prefix name *pn* of the ontology, variables x and y, and active voice a
Output: A SPARQL construct that represents R

if *a is true* **then**
  return `?x pn:R ?y`
else
  return `?y pn:R ?x`

Algorithm 6: *intersect*

Input : Two classes `C` and `D` represented in SPARQL
Output: The intersection of `C` and `D`

return `C . D`

noted by the child of `that-node`. Therefore, Algorithm 4 calls *transform* with `and-node` as input, to obtain a SPARQL description for the conjunction.

Algorithm 7: *transformWithNode*

Input : A with-node n, the prefix name *pn* of the ontology, a variable x
Output: A SPARQL construct Q that represents the concept in n

if *n.noun is functional* **then**
  if *n.noun is boolean with value true* **then**
    // e.g., [references] with clinical study
    $Q \leftarrow dataProp(x, pn, n.noun, 'true')$;
  else if *n.noun is boolean with value false* **then**
    // e.g., [references] without clinical study
    $Q \leftarrow dataProp(x, pn, n.noun, 'true')$;
    $Q \leftarrow complement(Q)$;
  else
    //*n.noun* is nonboolean
    if *n.operator is '='* **then**
      // e.g., [movements] with actuation =
      // 'series elastic'
      $Q \leftarrow dataProp(x, pn, n.noun, n.noun.value)$;
    else if *n.operator is '! ='* **then**
      $Q \leftarrow dataProp(x, pn, n.noun, n.noun.value)$;
      $Q \leftarrow complement(Q)$;
    else
      // e.g., [movements] with active
      // DOF ≥ 2
      $y \leftarrow$ generate a new variable;
      $Q \leftarrow dataProp(x, pn, n.noun, y)$;
      $F \leftarrow filter(y, n.operator, n.non.value)$;
      $Q \leftarrow intersect(Q, F)$;
else
  if $n.quantifier = SOME$ **then**
    // e.g., [movements] with some
    // transmission
    $y \leftarrow$ generate a new variable;
    $Q \leftarrow dataProp(x, pn, n.noun, y)$;
  else if $n.quantifier = ALL$ **then**
    // e.g., [robots] with all targeted disorders
    foreach *value* $v \in n.noun.range$ **do**
      $Q_1 \leftarrow dataProp(x, pn, n.noun, v)$;
      $Q \leftarrow intersect(Q, Q_1)$;
  else
    // e.g., [references] with no URL
    $y \leftarrow$ generate a new variable;
    $Q \leftarrow dataProp(x, pn, n.noun, y)$;
    $Q \leftarrow complement(Q)$;
return Q

Algorithm 8: *dataProp*

Input : A data property *R* of a class, an instance *a* denoting a possible value of *R*, the prefix name *pn* of the ontology, variable *x*
Output: A SPARQL construct that represents the value restriction of *R* to *a*

return `?x pn:R 'a'`

Algorithm 9: *union*

Input : Two classes `C` and `D` in SPARQL
Output: The union of `C` and `D`

return `{C} UNION {D}`

The first child of `and-node` describes the non-boolean data property (i.e., with actuation='electrical') expressed as an equality. By *transformWithNode* (Algorithm 7), which further calls *dataProp* (Algorithm 8), it is transformed into a SPARQL expression as follows:

```
?y rr:has_Actuation 'electrical'
```

The second child of `and-node` contains a simple disjunction (i.e., with transmission = 'cable drive' or with transmission='direct drive'). The algorithm transforms the information in the children of `or-node` into value restrictions and disjoins them (Algorithm 9):

```
{?y rr:has_Transmission
              'cable drive'.}
UNION
{?y rr:has_Transmission
              'direct drive'.}
```

Then the two children of `and-node` are conjoined first with each other (Algorithm 6), and then with the description of the object property above. This resulting concept is returned from Algorithm 4 (and thus from Algorithm 3), and then it is combined with the class description above as follows:

```
?x rdf:type rr:RehabRobots.
?x rr:targets ?y.
?y rdf:type rr:ShoulderMovements.
?y rr:has_Actuation 'electrical'.
{?y rr:has_Transmission
              'cable drive'.}
UNION
{?y rr:has_Transmission
              'direct drive'.}
```

After a SPARQL construct is obtained by Algorithm 1, a SPARQL query is formed as follows. We start with a PREFIX part and we declare the name space (the location of an ontology on the Web) of REHABROBO-ONTO. Next, we continue with a SELECT clause. The instances of type `RehabRobots`, by themselves, are not meaningful to the users. Thus, we want to display the names of the instances to the users. We specify it with an additional triple in the beginning of the WHERE clause, and continue the clause with the transformed SPARQL construct:

```
PREFIX rdf: <http://www.w3.org/...>
PREFIX rr:
   <http://www.semanticweb.org/...>

SELECT DISTINCT ?name
WHERE {
  ?x rr:has_Name ?name.
  ?x rdf:type rr:RehabRobots.
  ?x rr:targets ?y.
  ?y rdf:type rr:ShoulderMovements.
  ?y rr:has_Actuation 'electrical'.
  {?y rr:has_Transmission
     'cable drive'.}
  UNION
  {?y rr:has_Transmission
     'direct drive'.}
}
```

Example 2. Let us now describe some other aspects of the transformation by another example. Consider, for instance, the transformation of Q3 presented in Figure 7. This query is about movement quality metrics `?m` and the relative clause involves universal quantification over all robots with motion capability = 'grounded' that has assessment `?m`. Therefore, the main algorithm starts with generating the appropriate class `MovementQualityAssessment` for `root-node`:

```
?m rdf:type ?a.
?a rdfs:subClassOf
      rr:MovementQualityAssessment.
```

and then transforms `that-node` into SPARQL by calling Algorithm 3, which further calls Algorithm 10, as follows:

Algorithm 10: *forall*

Input : An object property R (i.e., *n.verb*) associated with the class of a that/with-node (i.e., *n.type*), a class C (i.e., *n.verb.type*), a prefix name *pn* of the ontology, a variable x, a child node m (i.e., *n.child* – possibly null)
Output: A SPARQL construct that represents the universal restriction of C relative to R

// e.g., [robots that target] all shoulder movements
// *n.type* is "robots", R is "targets",
// C is "shoulder movements"

$y \leftarrow$ generate a new variable;
$Q_1 \leftarrow$ *transformClass*(y, pn, C);
$Q_2 \leftarrow$ *objProp*$(x, pn, R, y, n.active)$;
$Q_3 \leftarrow$ *intersect*(Q_1, Q_2);
if *m is not null* **then**
  $Q_3 \leftarrow$ *intersect*$(Q_3,$ *transform*$(m, pn, y))$;

$z \leftarrow$ generate a new variable;
$Q_4 \leftarrow$ *transformClass*(z, pn, C);
if *m is not null* **then**
  $Q_4 \leftarrow$ *intersect*$(Q_4,$ *transform*$(m, pn, z))$;

$Q_5 \leftarrow$ *objProp*$(x, pn, R, z, n.active)$;
$Q_5 \leftarrow$ *complement*(Q_5);

$Q \leftarrow$ *complement*$($*intersect*$(Q_4, Q_5))$;

return Q

// the output for the example above:
// `?y rdf:type pn:C .`
// `?x pn:R ?y .`
// `FILTER NOT EXISTS {`
// `  ?y' rdf:type pn:C .`
// `  FILTER NOT EXISTS {`
// `    ?x pn:R ?y' }}`

Algorithm 11: *complement*

Input : A class C described in SPARQL
Output: The complement of C

return `FILTER NOT EXISTS {C}`

```
?r rdf:type rr:RehabRobots.
?r rr:hasMotionCapability 'grounded'.
?r rr:hasAssessment ?m.
FILTER NOT EXISTS {
  ?r2 rdf:type rr:RehabRobots.
  ?r2 rr:hasMotionCapability
                        'grounded'.
  FILTER NOT EXISTS {
      ?r2 rr:hasAssessment ?m. } }
```

Algorithm 12: *filter*

Input : A variable x, a comparison operator *op*, a prefix name *pn* of the ontology, and a constant value *val*
Output: A test

return `FILTER (?x pn:op val)`

A universal restriction $\forall x A(x)$ corresponds to a negated existential restriction $\neg\exists x \neg A(x)$. Each `FILTER NOT EXISTS` expression above (generated by Algorithm 11) corresponds to a negation. The outer `FILTER NOT EXISTS` describes that there exists no robot `?r2` with motion capability 'grounded' that has not evaluated `?m`.

Example 3. Let us consider the transformation of Q5 presented in Figure 8. This query is about publications `?p` (described by their titles `?t`) and involves a negative condition about these publications, i.e., "that do not reference any robots [with ...]". This negative condition is represented by a `FILTER NOT EXISTS` expression. The query also involves a data property check with an inequality: "with active degree of freedom $\leq$ 1". This check is performed using a `FILTER` operator (Algorithm 12).

6. Answering Queries Using Pellet

We use the DL reasoner PELLET to find answers to queries, through the Jena framework. Consider, for instance, the query

> What are the robots that target some wrist movements with actuation='series elastic'?

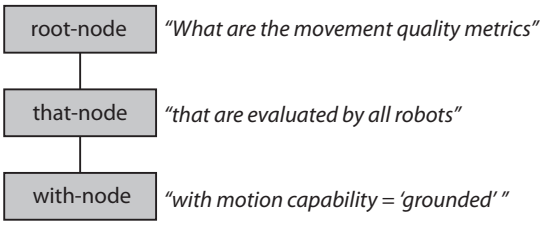
whose SPARQL representation is as follows.

```
SELECT DISTINCT ?name
WHERE {
  ?robot1 rr:has_Name ?name.
  ?robot1 rdf:type rr:RehabRobots.
  ?robot1 rr:targets ?movement1.
  ?movement1 rdf:type
                rr:WristMovements.
  ?movement1 rr:has_Actuation
     'series elastic'.
}
```

After loading REHABROBO-ONTO into PELLET, we present this SPARQL query to PELLET, and get the following answer:

Q3. What are the movement quality metrics that are evaluated by all robots with motion capability = 'grounded' ?

Q3 in Query Description Tree:



Q3 in SPARQL:

```
SELECT DISTINCT ?m
WHERE {
  ?m rdf:type ?a.
  ?a rdfs:subClassOf rr:MovementQualityAssessment.
  ?r rdf:type rr:RehabRobots.
  ?r rr:has_Motion_Capability 'grounded'.
  ?r rr:hasAssessment ?m.
  FILTER NOT EXISTS {
    ?r2 rdf:type rr:RehabRobots.
    ?r2 rr:hasMotionCapability 'grounded'.
    FILTER NOT EXISTS {
      ?r2 rr:hasAssessment ?m.
    } }
}
```

Fig. 7. Transformation of a query, Q3, from REHABROBO-CNL to a SPARQL.

```
( ?name = "AssistOn-Mobile" )
```

Consider, for instance, another query:

> What are the publications with place of publication 'ICORR' and that reference some robots that are owned/maintained by some users with institution 'Sabanci University' ?

The SPARQL representation of this query is as follows.

```
SELECT DISTINCT ?name
WHERE {
  ?publication1 rr:has_Title ?name.
  ?publication1 rdf:type
                rr:References.
  ?publication1 rr:has_PublishedAt
                          'ICORR'.
  ?robot1 rr:hasReference
                    ?publication1.
  ?robot1 rdf:type rr:RehabRobots.
  ?robot1 rr:ownedBy ?user1.
  ?user1 rdf:type rr:Owners.
  ?user1 rr:has_Institution
             'Sabanci University'.
}
```

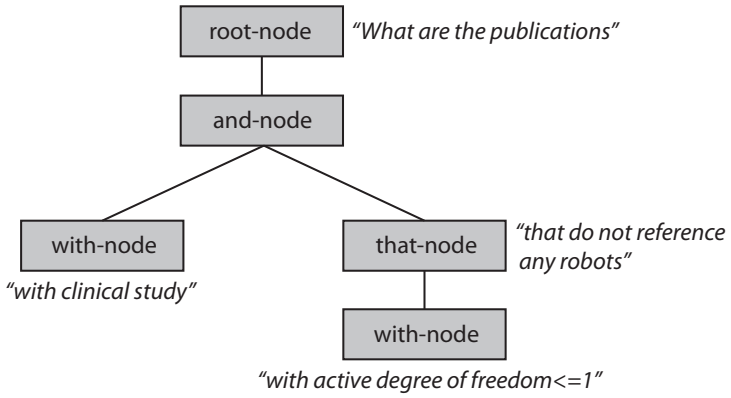
We get the following answers from PELLET to this query:

```
( ?name = "Brain Computer Interface
  based Robotic Rehabilitation with
  Online Modification of Task Speed" )

( ?name = "Passive Velocity Field
  Control of a Forearm-Wrist
  Rehabilitation Robot" )
( ?name = "Design of a reconfigurable
  ankle rehabilitation robot and its
  use for the estimation of the ankle
  impedance" )
```

Q5. What are the publications with clinical study and that do not reference any robots with active degree of freedom ≤ 1?

Q5 in Query Description Tree:



Q5 in SPARQL:

```
SELECT DISTINCT ?t
WHERE {
  ?p rdf:type rr:References.
  ?p rr:has_Title ?t.
  ?p rr:has_Clinical_Study 'true'^^xsd:boolean.
  FILTER NOT EXISTS {
    ?r rr:hasReference ?p.
    ?r rdf:type rr:RehabRobots.
    ?r rr:has_Active_DOF ?val1. FILTER(?val1<=(1))
  }
}
```

Fig. 8. Transformation of a query, Q5, from REHABROBO-CNL to a SPARQL.

7. Evaluations of REHABROBO-QUERY

We evaluated the underlying methods of the system REHABROBO-QUERY empirically over a variety of queries, and its user-interface by means of a survey with the help of participants of different expertise.

7.1. Experimental evaluations

We experimentally evaluated the computational performance of REHABROBO-CNL to SPARQL transformation and query answering. We identified 20 queries (Table 8) of different types as discussed in Section 2, paying also attention to their structures (e.g., conjunctive, disjunctive, negated, quantified, nested). Most of these queries were constructed based on the feedback of the experts, and evaluated as useful queries. We also constructed a few queries with different structures supported by REHABROBO-CNL, for the purpose of evaluating REHABROBO-QUERY. In particular, Q6 was constructed to illustrate the use of quantifiers, whereas Q19 was constructed to illustrate the use of nested conjunctions/disjunctions. These two queries are relevant for evaluation, but experts found Q6 not specific enough and Q19 too specific to be useful.

All computations were performed using a T1 Micro Instance of Amazon EC2. During the experiments, a T1 Micro Instance operated with up to 600 MB memory and two EC2 compute units, each unit providing an equivalent CPU capacity of a 1.0–1.2 GHz 2007 Xeon

processor. For query answering, we used PELLET version 2.3.0.

Experimental results in Table 8 indicate that the transformation from REHABROBO-CNL to SPARQL can be completed within milliseconds, while query answering can take up to 2 seconds. Please recall that REHABROBO-CNL to SPARQL transformation is linear time in the size of the query and thus it is independent of the population level of REHABROBO-ONTO while SPARQL query answering is a harder problem [38] and it is likely to be adversely affected as REHABROBO-ONTO is further populated. However, REHABROBO-QUERY has been implemented in Amazon EC2 to ensure scalability of the system even after increasing the size of REHABROBO-ONTO. In particular, larger EC2 instances with more memory and CPU units can be recruited as the ontology is further populated and increases in size.

7.2. User evaluations

We evaluated the usefulness of REHABROBO-QUERY with a group of real users to demonstrate how the proposed approach fits different needs of people working in the interdisciplinary domain of rehabilitation robotics. For that, we conducted a survey with 15 experts working in this domain. These experts included clinicians from the Rehabilitation Institute of Chicago and Acıbadem Hospital at İstanbul, rehabilitation robotics engineers from Northwestern University and Sabancı University, and experts from an SME specializing in commercialization of rehabilitation robots.

Each participant was given a brief overview of REHABROBO-QUERY (i.e., the variety of knowledge about rehabilitation robotics that is represented at REHABROBO-ONTO, the types of queries supported by REHABROBO-QUERY to access this knowledge) and a direct access to the interface. Participants were asked to explore the system by freely formulating several queries of each type with respect to REHABROBO-CNL, about mechanical aspects of robots, joint movements, evaluation metrics, and relevant publications. After exploring REHABROBO-QUERY with different queries of their choice, participants were asked to fill in a questionnaire. The questions that were presented to the participants and the analysis of responses to this questionnaire are presented in Table 9. In addition to the questions in Table 9, the questionnaire included an open-ended question that asks for general evaluations and feedback of the participants.

The statistical analysis of responses revealed that occupation as a factor was not statistically significant at the 0.05 level for any of the survey questions; hence, all responses were aggregated for reporting. The Cronbach's α values were calculated for the whole survey, and the α value was evaluated to be greater than 0.7, indicating high reliability of the survey.

The survey included 3 quantifiable questions: q1 aimed at evaluating the usefulness of several query types supported by REHABROBO-QUERY, q2 was for assessing the potential use of REHABROBO-QUERY, and q3 was for determination of target population of REHABROBO-QUERY. For all questions, the five-point Likert scale, ranging from "1" (e.g., least useful) to "5" (e.g., most useful) was used.

The main results of the survey can be summarized as follows:

- Responses to q1 indicate that participants *agreed* with the usefulness of REHABROBO-QUERY while querying about mechanical properties of robots, joint movements targeted by rehabilitation robots, evaluation metrics for rehabilitation robots, and publications about rehabilitation robots.
- From q2, we can infer that participants found REHABROBO-QUERY useful for identifying relevant rehabilitation robots suitable for a therapy, identifying researchers/labs for relevant rehabilitation robots, and identifying outcomes of clinical studies for relevant rehabilitation robots.
- Responses to q3 indicate that participants evaluated REHABROBO-QUERY to be highly useful for rehabilitation robotics engineers, students/researchers and physical medicine experts, while also finding it useful for hospitals.

While the participants were testing REHABROBO-QUERY, we made ourselves available for any questions they might have. All participants were able to complete several examples of each query type and no participant required any clarifications about query construction during testing, even though the participants were not familiar with REHABROBO-QUERY before. Furthermore, we interviewed the participants after they completed their evaluations. In these interviews and the open-ended question in the questionnaire, we did not receive any negative feedback about query construction. These experiences can be viewed as a positive evidence that supports clarity of REHABROBO-CNL

Table 8
Experimental Evaluation of Transformation and Query Answering

	Query	REHABROBO-CNL to SPARQL [sec]	Query Answering [sec]
Q1	What are the robots that target shoulder movements and that have at least 210° RoM for the flexion/extension movements of the shoulder?	0.0002	0.2277
Q2	What are the robots that target wrist movements and that have at least 2 active degrees of freedom?	0.0002	0.2083
Q3	What are the shoulder robots that target flexion/extension movements and that do not target elevation/depression movements?	0.0002	0.6580
Q4	What are the robots with active degree of freedom $\geq$ '3' and that target all pelvic girdle movements with backdrivability type = 'passive'?	0.2658	0.3479
Q5	What are the ankle robots that target movements with electrical actuation and with cable drive transmission?	0.0003	0.1955
Q6	What are the movements that are targeted by some robots with (some intervention time or with all targeted disorders)?	0.0002	0.5368
Q7	What are the movements that are targeted by the robot 'AssistOn-Finger' and with minimum range of motion $\geq$ '20'?	0.0002	1.8593
Q8	What are the movements that are not targeted by any robots with kinematic type = 'redundant' and with mechanism type $\neq$ 'parallel'?	0.0002	0.3790
Q9	What are the effort metrics that are evaluated by some robots with active degree of freedom $\geq$ 2?	0.0002	0.2765
Q10	What are the movement quality metrics that are evaluated by all robots with motion capability = 'grounded'?	0.0003	0.2687
Q11	What are the kinematic aspect metrics that are evaluated by some robots that target all elbow movements?	0.1569	0.4238
Q12	What are the muscle strength metrics that are evaluated by robots that target all wrist movements with transmission = 'direct drive'?	0.1570	0.3465
Q13	What are the psychomotoric aspect metrics that are evaluated by all robots with kinematic type = 'redundant' and that target all ankle movements with minimum range of motion $\geq$ '30'?	0.1646	0.2351
Q14	What are the publications with clinical study and that do not reference any robots with active degree of freedom $\geq$ 1?	0.0003	0.1345
Q15	What are the publications without clinical study or that reference some robots that do not evaluate any movement quality metrics?	0.0002	0.1474
Q16	What are the publications with place of publication 'ICORR' and that reference some robots that are owned/maintained by some users with institution 'Sabanci University'?	0.0003	0.1495
Q17	What are the users that own/maintain some robots that target all ankle movements?	0.1392	0.1148
Q18	What are the robots that target some wrist movements with actuation='series elastic'?	0.0004	0.1563
Q19	What are the robots with no targeted disorder or (with intervention time!='chronic' and with motion capability='grounded') or with no disorder level?	0.0003	0.1432
Q20	What are the robots with interaction type = 'exoskeleton' and that target some finger movements (with actuation = 'electrical' or with actuation = 'hydraulic' or with actuation = 'series elastic')?	0.0003	0.1725

Table 9
Survey Questions and Summary Statistics

q1: Please rate the usefulness of the following aspects of REHABROBO-QUERY while querying about		
	Mean	σ^2
	4.08	0.70
Mechanical properties of robots	4.07	0.80
Joint movements targeted by rehabilitation robots	4.00	0.65
Evaluation metrics for rehabilitation robots	3.93	0.70
Publications about rehabilitation robots	4.33	0.62
q2: For what purpose would you use REHABROBO-QUERY in your research?		
	Mean	σ^2
	4.18	0.81
Identifying relevant rehabilitation robots suitable for a therapy	4.40	0.83
Identifying researchers/labs for relevant rehabilitation robots	4.00	0.65
Identifying outcomes of clinical studies for relevant rehabilitation robots	4.13	0.91
q3: Overall how would you rate the usefulness of REHABROBO-QUERY for the following user groups?		
	Mean	σ^2
	4.33	0.66
Rehabilitation robotics engineers	4.67	0.49
Physical medicine experts	4.14	0.66
Hospitals	3.92	0.76
Students/researchers	4.53	0.52

(e.g., participants were able to construct their queries using REHABROBO-QUERY) and the easy use of the interface (e.g., participants had no difficulty finding relevant tabs/menus, thanks to its interactive and intelligent design).

We received one improvement suggestion about the interface: while formulating queries one after another, it would be good to re-use some parts of the previous queries. Similarly, two of the participants noted that complex negative queries (especially the ones with double negations) can become unnatural and hard to understand. Relevant updates are planned as part of our future work.

8. Related Work

The most related work involves ontology systems and tools that support natural language queries over ontologies.

Development of natural language interfaces that provide query answering over ontologies has been subject of research for many years. For this reason, many systems [4,6,8,22,26,27,31,33,45,48,50] have been developed that propose various approaches over some common challenges, such as processing of the natural language input (balancing ambiguity and expressiveness) and support for broad or narrow domains (portability).

One of the most recently developed systems is BIOQUERY-ASP [18], which is a software system that answers natural language queries over biomedical databases and ontologies related to drug discovery. It utilizes Answer Set Programming (ASP) [32] to query such knowledge resources. It allows the users to enter queries in a controlled natural language from its user interface, and then answers the queries by transforming the query in a controlled natural language into an ASP program. To enable interoperability over multiple biomedical ontologies and databases, it integrates ontologies via a rule layer in ASP. To answer queries, it utilizes ASP solvers such as CLASP [23] and CLASP-NK, and it also provides explanations to the queries. BIOQUERY-ASP covers sophisticated queries with nested relative clauses, aggregates, superlatives, nonmonotonic negation, etc. as necessitated by its domain of drug discovery; and to represent and answer these queries, ASP provides a better frame-

work. REHABROBO-QUERY covers simpler queries due to the requirements of its domain of rehabilitation robotics, and thus sophisticated aspects of ASP are not needed. W3C recommended Semantic Web technologies provide a sufficiently good framework to represent and answer these queries. Both systems provide an interactive and intelligent user-interface to construct queries with respect to their own CNLs, avoiding ambiguities.

Ferrández et al. [21] introduce QACID, which covers a movie ontology. The idea is to train the system using many queries and keep the resulting set of clusters (mostly asked questions) in a database. Then, manually, each query type is associated with a SPARQL query. Finally, the queries are answered by a query engine that is implemented in QACID and proposed as a new entailment-based engine. Although there is no formal description of what types of queries are supported by QACID, all of the examples given in the paper are simple queries that do not involve disjunctions, negations, (nested) relative clauses, and quantifiers. In that sense, REHABROBO-QUERY supports a greater variety of queries. In terms of user interface, since QACID allows the users to enter queries as free text, there may be ambiguities. REHABROBO-QUERY provides an interactive user-interface to enter queries, and thus prevents ambiguities in the query.

FREYA [10] is developed by the creators of and as a development upon QUESTIO [45]. In order to support natural language queries, it uses Stanford Parser [12] to generate a parse tree. Then, using GATE libraries [9], it tries to find some ontology concepts that can be mapped to the query terms. Then, it generates a SPARQL query and executes the query using the inference engine in BIGOWLIM, that supports SPARQL, on top of Sesame. It relies on clarification dialogues with users in the cases of ambiguity or in the cases where the system cannot find an answer to a query. Over time, the system learns to ask the correct questions to the users by placing correct suggestions on top of similar queries. The system is tested on one dataset, and it is stated that FREYA failed to answer some questions (e.g., queries including negation) correctly. These questions could not be mapped to a SPARQL query in spite of the clarification dialogues and the learning mechanism. REHABROBO-QUERY supports queries with negation, and can guide the user to construct queries according to its CNL without any ambiguities.

Lopez et al. [34] introduce PowerAqua, which is evolved from AquaLog [33]. It provides natural language querying over multiple ontologies; thus, supports high scalability and portability. It uses GATE libraries and WordNet [20] to process natural language queries. It transforms the queries to triples and answers them with its own query engine. To limit the search space, it uses filtering and ranking heuristics. Since it does not contain any linguistic knowledge in the background, it has a limited linguistic coverage. It is good at answering simple questions yet it fails on questions that contain comparisons and quantifiers. REHABROBO-QUERY supports queries with quantifiers. In terms of user interface, PowerAqua queries are entered as free text, while some examples are illustrated to the user if requested. In that sense, there may be ambiguities of queries. Due to its CNL-based interactive interface, REHABROBO-QUERY prevents ambiguous queries.

Valencia-García et al. [46] introduce OWLPath, which gets user queries in a controlled natural language, transforms it into a SPARQL query and executes the query over an ontology via Jena framework and using the DL reasoner PELLET. The statements in its CNL start with "View any..." and follow English grammar. However, they are not full and valid English sentences. Although it is stated that OWLPath provides a Web interface through AJAX, it is not available online. OWLPath provides suggestions for queries, relative to the terminology of its CNL; in that sense, REHABROBO-QUERY and OWLPath have some similarities. For each condition in the query, OWLPath adds a FILTER statement in the SPARQL query. Therefore, the transformation of the query into SPARQL is not, in fact, a transformation to triples but a set of FILTER statements. In that sense, the underlying transformation of REHABROBO-QUERY is more systematic in covering other types of queries (e.g., with negation).

9. Conclusion

In this article, we have introduced methods for representing queries about rehabilitation robotics in a controlled natural language, transforming them into formal queries, and computing answers to them using automated reasoners over the first formal rehabilitation robotics ontology REHABROBO-ONTO. We have also introduced an interactive and intelligent user-interface as part of the software system REHABROBO-QUERY, that guides the users during this whole process in such a way that the users do not have to know about the underlying logical formalism of the ontology or the for-

malism to represent queries; and they do not have to know about the use of the technologies for computing answers to their questions.

By means of answering sophisticated queries over REHABROBO-ONTO, appropriate rehabilitation robots for a particular patient or a physical therapy can be found or designed; this further paves the way for translational physical medicine (from bench-to-bed and back) and personalized physical medicine. Also, REHABROBO-QUERY aids the exchange of information across rehabilitation robots researchers over the world, and therefore improves the state-of-the-art; it allows to identify "gaps" in functionality of rehabilitation robots, that can further improve research efforts.

Having a structured formal representation of knowledge about rehabilitation robotics as an ontology, allows answering complex queries that require integration with other knowledge resources (e.g., patient databases, disease ontologies). Along this research direction, integration of REHABROBO-ONTO with existing anatomy, disease and patient ontologies can be achieved by providing a rule layer between these ontologies and REHABROBO-ONTO, for integration of the related concepts [17]. In addition, some extensions in the grammar of REHABROBO-CNL, the algorithms and the user interface of REHABROBO-QUERY are needed to be able to answer complex queries about therapies, diseases and anatomy. These studies concerning interoperability of REHABROBO-ONTO is part of our ongoing work. Further user studies are also planned as part of our future work.

Acknowledgment

We would like to thank Sibel Aksu Yildirim and Muhammed Kilinc (Hacettepe University, School of Physical Therapy and Rehabilitation) for their feedbacks on the types of queries from the perspectives of physical medicine, and Agis Papantoniou (Cognizone) for his help with the design of REHABROBO-QUERY. We would like to thank Ahmetcan Erdogan (Rehabilitation Institute of Chicago), Mustafa Yalcin, Murat Isik and Ata Otaran (Human-Machine Interaction Lab, Sabanci University), and Ezgi Demirel (Knowledge Representation and Reasoning Group, Sabanci University) for their helps with testing REHABROBO-QUERY. We would like to thank members of European Network on Robotics for NeuroRehabilitation for useful discussions. We are also very grateful for the participants of our survey about REHABROBO-QUERY.

We also would like to thank the reviewers Nick Bassiliades, Antonis Bikakis, and Eleni Kamateri for their insightful questions and useful suggestions on earlier drafts to improve the presentation of our work.

References

[1] Global burden of stroke. http://www.who.int/cardiovascular_diseases/en/cvd_atlas_15_burden_stroke.pdf.

[2] Economic cost of stroke. http://www.who.int/cardiovascular_diseases/en/cvd_atlas_17_economics.pdf.

[3] G. Antoniou and F. V. Harmelen. Web ontology language: OWL. In *Handbook on Ontologies in Information Systems*, pages 67–92, Springer, 2003. https://doi.org/10.1007/978-3-540-24750-0_4.

[4] A. D. L. Battista, N. Villanueva-Rosales, M. Palenychka, and M. Dumontier. SMART: A web-based, ontology-driven, semantic web query answering application. In *Semantic Web Challenge*, volume 295, 2007. http://ceur-ws.org/Vol-295/paper17.pdf.

[5] N. Bayona, K. Bitensky J.and Salter, and R. Teasell. The role of task-specific training in rehabilitation therapies. *Topics in Stroke Rehabilitation*, 12(3):58–65, 2005. https://doi.org/10.1310/BQM5-6YGB-MVJ5-WVCR.

[6] A. Bernstein and E. Kaufmann. GINO - a guided input natural language ontology editor. In I. Cruz et al. (eds), *Proceedings of the 5th International Semantic Web Conference (ISWC)*, pages 144–157, Springer, 2006. https://doi.org/10.1007/11926078_11.

[7] H. D. P. Butefisch, C.and Hummelsheim and K. Mauritz. Repetitive training of isolated movements improves the outcome of motor rehabilitation of the centrally paretic hand. *Journal of the Neurological Sciences*, 130(1):59–68, 1995. https://doi.org/10.1016/0022-510X(95)00003-K.

[8] P. Cimiano, P. Haase, J. Heizmann, M. Mantel, and R. Studer. Towards portable natural language interfaces to knowledge bases - the case of the ORAKEL system. *Data and Knowledge Engineering*, 65(2)(2):325–354, 2008. https://doi.org/10.1016/j.datak.2007.10.007.

[9] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6)*, 2011. https://gate.ac.uk/releases/gate-6.1-build3913-ALL/tao.pdf.

[10] D. Damljanovic, M. Agatonovic, and H. Cunningham. FREyA: an interactive way of querying linked data using natural language. In R. Garcia-Castro, D. Fensel, G. Antoniou (eds), *Proceedings of the 8th International Semantic Web Conference (ISWC)*, pages 125–138, Springer, 2012. https://doi.org/10.1007/978-3-642-25953-1_11.

[11] B. D'Arcus and F. Giasson. Bibliographic ontology specification. *Madrid: Biblioteca Nacional Española*, 2009. http://bibliontology.com/.

[12] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure trees.

In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006. https://nlp.stanford.edu/pubs/LREC06_dependencies.pdf.

[13] Z. Dogmus, G. Gezici, V. Patoglu, and E. Erdem. Developing and maintaining an ontology for rehabilitation robotics. In *Proceedings of International Conference on Knowledge Engineering and Ontology Development (KEOD)*, pages 389–395, 2012. https://doi.org/10.5220/0004145303890395.

[14] Z. Dogmus, A. Papantoniou, M. Kilinc, S. A. Yildirim, E. Erdem, and V. Patoglu. Rehabilitation robotics ontology on the cloud. In *Proceedings of IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, 2013. https://doi.org/10.1109/ICORR.2013.6650415.

[15] Z. Dogmus, V. Patoglu, and E. Erdem. Answering natural language queries about rehabilitation robotics ontology on the cloud. In *Proceedings of International Conference on Knowledge Engineering and Ontology Development (KEOD)*, pages 75–83, 2014. https://doi.org/10.5220/0005081400750083.

[16] Z. Dogmus, E. Erdem, and V. Patoglu. RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud. *Robotics and Computer-Integrated Manufacturing*, 33:100–109, 2015. https://doi.org/10.1016/j.rcim.2014.08.010.

[17] Z. Dogmus, V. Patoglu, and E. Erdem. Interoperability of RehabRobo-Onto. *PeerJ Preprints*, 5:e3255v2, 2017. https://peerj.com/preprints/3255.pdf.

[18] E. Erdem, H. Erdogan, and U. Oztok. BIOQUERY-ASP: Querying biomedical ontologies using answer set programming. *EMBnet.journal*, 18(B):62–64, 2011. https://doi.org/10.14806/ej.18.B.551.

[19] I. H. Ertas, E. Hocaoglu, and V. Patoglu. AssistOn-Finger: An under-actuated finger exoskeleton for robot-assisted tendon therapy. *Robotica*, 32(8):1363–1382, 2014. https://doi.org/10.1017/S0263574714001957.

[20] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. Mit Press, 1998.

[21] O. Ferrández, R. Izquierdo, S. Ferrández, and J. L. Vicedo. Addressing ontology-based question answering with collections of user queries. *Information Processing and Management*, 45 (2):175 – 188, 2009. https://doi.org/10.1016/j.ipm.2008.09.001.

[22] A. Frank, H.-U. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer. Question answering from structured knowledge sources. *Journal of Applied Logic*, 5(1):20 – 48, 2007. https://doi.org/10.1016/j.jal.2005.12.006.

[23] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A conflict-driven answer set solver. In C. Baral, G. Brewka, J. Schlipf (eds), *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 260–265, Springer, 2007. https://doi.org/10.1007/978-3-540-72200-7_23.

[24] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003. https://doi.org/10.1016/S1071-5819(02)00127-1.

[25] I. Horrocks, P. F. Patel-Schneider, and F. V. Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1:2003, 2003. https://doi.org/10.1016/j.websem.2003.07.001.

[26] E. Kaufmann, A. Bernstein, and R. Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proceedings of the 5th International Semantic Web Conference (ISWC)*, pages 980–981. Springer, 2006.

[27] E. Kaufmann, A. Bernstein, and L. Fischer. NLP-Reduce: A naive but domain-independent natural language interface for querying ontologies. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, 2007.

[28] T. Kuhn. A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1):121–170, MIT Press Cambridge, 2014. http://dx.doi.org/10.1162/COLI_a_00168.

[29] G. Kwakkel, R. Wagenaar, J. Twisk, G. Langkhorst, and J. Koetsier. Intensity of leg and arm training after primary middle-cerebral artery stroke: A randomized trial. *Lancet*, 35:191–196, 1999. http://dx.doi.org/10.1016/S0140-6736(98)09477-X.

[30] G. Kwakkel, B. J. Kollen, and H. I. Krebs. Effects of Robot-Assisted Therapy on Upper Limb Recovery After Stroke: A Systematic Review. *Neurorehabilitation and Neural Repair*, 22:111–121, 2008. https://doi.org/10.1177%2F1545968307305457.

[31] Y. Lei, V. Uren, and E. Motta. SemSearch: A search engine for the semantic web. In S. Staab, V. Svatek (eds), *Proceedings of 5th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, Lect. Notes in Comp. Sci.*, pages 238–245. Springer-Verlag, 2006. https://doi.org/10.1007/11891451_22.

[32] V. Lifschitz. What is answer set programming?. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, volume 3, pages 1594–1597, 2008. http://dl.acm.org/citation.cfm?id=1620270.1620340.

[33] V. Lopez, V. Uren, E. Motta, and M. Pasin. AquaLog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):72–105, 2007. http://dx.doi.org/10.1016/j.websem.2007.03.003.

[34] V. Lopez, M. Fernández, E. Motta, and N. Stieler. PowerAqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012. https://doi.org/10.3233/SW-2011-0030.

[35] J. M. McDowd, D. L. Filion, P. S. Pohl, L. G. Richards, and W. Stiers. Attentional abilities and functional outcomes following stroke. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 58(1):45–53, 2003. https://doi.org/10.1093/geronb/58.1.P45.

[36] H. Munawar, M. Yalcin, and V. Patoglu. Redundant kinematics and workspace centering control of assiston-gait overground gait and balance trainer. In *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3704–3710, 2016. https://doi.org/10.1109/ICRA.2016.7487556.

[37] K. Nykanen. The effectiveness of robot-aided upper limb therapy in stroke rehabilitation: A systematic review of randomized controlled studies. Master's thesis, Jyvaskyla University, 2010. https://jyx.jyu.fi/handle/123456789/23001.

[38] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. In I. Cruz et al. (eds), *Proceedings of the 5th International Semantic Web Conference (ISWC)*, pages 30–43, Springer, 2006. https://doi.org/10.1007/11926078_3.

[39] S. Peroni and D. Shotton. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17 (Supplement C):33 – 43, 2012. https://doi.org/10.1016/j.websem.2012.08.001.

[40] T. Platz. Evidence-based arm rehabilitation – A systematic review of the literature. *Nervenartz*, 74(10): 841–849, 2003. https://doi.org/10.1007/s00115-003-1549-7.

[41] G. B. Prange, M. J. Jannink, C. G. Groothuis-Oudshoorn, H. J. Hermens, and M. J. Ijzerman. Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. *Journal of Rehabilitation Research and Development*, 43:171–184, 2006. https://doi.org/10.1682/JRRD.2005.04.0076.

[42] E. Prud'Hommeaux, A. Seaborne, et al. SPARQL query language for RDF. *W3C recommendation*, 15, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[43] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53, 2007. https://doi.org/10.1016/j.websem.2007.03.004.

[44] A. Sunderland, D. Tinson, E. Bradley, D. Fletcher, H. Langton, and D. Wade. Enhanced physical therapy improves recovery of arm function after stroke: A randomised clinical trial. *Journal of Neurology, Neurosurgery and Psychiatry*, 55:530–535, 1992. http://dx.doi.org/10.1136/jnnp.57.7.856.

[45] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In S. Bechhofer, M. Hauswirth, J. Hoffmann, M. Koubarakis (eds), *Proceedings of the 5th European Semantic Web Conference (ESWC)*, pages 361–375, Springer, 2008. https://doi.org/10.1007/978-3-540-68234-9_28.

[46] R. Valencia-García, F. García-Sánchez, D. Castellanos-Nieves, and J. Fernández-Breis. OWLPath: An OWL ontology-guided query editor. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(1):121–136, 2011. https://doi.org/10.1109/TSMCA.2010.2048029.

[47] J. H. van der Lee, I. A. K. Snels, H. Beckerman, G. J. Lankhorst, R. J. Wagenaar, and L. M. Bouter. Exercise therapy for arm function in stroke patients: a systematic review of randomized controlled trials. *Clinical Rehabilitation*, 15(1):20–31, 2001. https://doi.org/10.1191/026921501677557755.

[48] C. Wang, M. Xiong, Q. Zhou, and Y. Yu. PANTO: A portable natural language interface to ontologies. In E. Franconi, M. Kifer, W. May (eds), *Proceedings of the 4th European Semantic Web Conference (ESWC)*, pages 473–487, Springer, 2007. https://doi.org/10.1007/978-3-540-72667-8_34.

[49] M. Yalcin and V. Patoglu. Kinematics and design of AssistOn-SE: A self-adjusting shoulder-elbow exoskeleton. In *Proceedings of 2012 4th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1579–1585, 2012. https://doi.org/10.1109/BioRob.2012.6290928.

[50] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu. SPARK: adapting keyword query to semantic search. In K. Aberer et al. (eds), *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC/ASWC)*, pages 694–707, Springer, 2007. https://doi.org/10.1007/978-3-540-76298-0_50.