D4.1: Natural Multimodal Interaction Baseline

Bernd Kiefer, Hans-Ulrich Krieger, Ivana Kruijff-Korbayová, Stefania Raccioppa

DFKI GmbH

⟨{kiefer,krieger,ivana.kruijff,stefania.racioppa}@dfki.de⟩

Project, project Id:	EU H2020 PAL / PHC-643783
Project start date:	March 1 2015 (48 months)
Due date of deliverable:	February 29 2016
Actual submission date:	some time
Lead partner:	DFKI
Revision:	final
Dissemination level:	PU

The present report describes the work carried out in the first project year regarding *Natural Multimodal Interaction*. It summarizes the Deliverable D4.1: "Natural Multimodal Interaction Baseline".

We defined a new formalism for the specification of dialogue policies that combines dialogue rules, knowledge representation and dialogue history in a unique way. We developed the first version of an ontology which specifies the data structures to be used by the dialogue specifications, dialogue history, and information state, and adapted our reasoning components, so that this knowledge source can be used efficiently once the formalism specification is fully implemented. We implemented a prototype for the experiments in year 1, which provide us with interaction data that can be used later to develop and evaluate the components in this work package.

1 Executive Summary

The current report summarises the results of WP4 for the first year of PAL. The overall objective of WP4 is to support the goals set for a patient using the PAL system by developing the means to conduct verbal communication, and to analyse textual data and extract relevant information. The components implemented in this work package must support this communication in a way to foster sustainable long-term interactions between a robot (or its avatar) and a human. This requires user-adaptive communication, coupling of verbal and non-verbal communication and grounding communication in long-term memory. In the first year, we have established a basis for verbal interaction processing corresponding to these objectives.

This document describes the work done to reach deliverable D4.1, *Natural multimodal interaction baseline*, and the relevant milestones that are comprised in it. Work has been carried out in the two currently active tasks of this work package, namely task T4.1 on *Robust interpretation of user input* and task T4.2 on *Flexible and adaptive multimodal generation*.

In task T4.1 we worked on a novel approach for dialogue management that treats the specification and storage of domain knowledge and interaction structure uniformly. This uniform knowledge layer will support almost all sub-tasks of WP4, such as natural language (NL) interpretation, NL generation, user and role adaptation, and the definition of dialogue policies. The modelling and storage of the user and domain information, and the interaction history is really a general topic that over-arches all work in WP4.

To provide this functionality, we developed an ontology that allows to represent interaction in a way that is natural and efficient to use. We heavily extended existing processing components, i.e., the reasoning engine and database layer, which make the data available to the interaction management and analysis. Besides, we connected the Google automatic speech recognition API to the common PAL infrastructure.

In Task T4.2 we adapted and extended an existing component for verbal generation of Italian using a deep-generation approach, involving utterance content planning and grammar-based surface realisation. Since the underlying system using this grammar is bi-directional, the grammar can later on also be used for interpretation. We also worked on a free Dutch text-to-speech voice for the Mary TTS system, in order to obtain a uniform voice for the virtual and the embodied robot.

We built a prototype system from existing components for the first set of experiments. The purpose of this prototype was to provide a system for data collection and experimentation in short time, before having the new system architecture in place. The unavoidable adaptations were kept to a minimum, since this prototype will not be sustained, but be replaced by the new, more powerful and flexible architecture and components. To inform the development of the system, we analysed the interaction data in order to evaluate how the children's perception of the robot is influenced by their individual interactions with the robot, and by specific relational verbal behaviours.

The results of Year 1 are presented in 5 published peer-reviewed conference papers, and two technical reports.

2 The role of *Multimodal Natural Interaction* in PAL

WP4 focuses on the actual multimodal interaction around mHealth-Apps and additional conversational functionality in support of the high-level targets set in WP2 and actions selected in WP3. The challenge is to produce natural, flexible, personalized interactions that are sustainable in th long term as well as allow to extract data about the user. To achieve this, we are incorporating findings from the literature about what aspects are important for long term engagement.

The processing challenges for this work package are the robustness of input interpretation, flexibility *and* personal adaptation of the generated output, handling different situational contexts for both the physical and graphical embodiment of PAL, and allowing for interactions with one child alone or in the presence of an audience of multiple children. Additional challenges are posed by the need for extendable thematic and linguistic coverage.

The core functional component developed in WP4 is a multimodal dialogue system with a repertoire of multimodal dialogue acts (combining verbal and non-verbal means) modulated by affect. Generation as well as interpretation will use parameterized dialogue acts as an interface schema to other modules to abstract away from specific aspects of, e.g., natural language or emotion expression. Based on the high-level targets from WP2, action selection from WP3, the dialogue state (including the latest child's input and interaction history) as well as a long-term memory, the multimodal output generation module decides which act to activate ("what to express") and how to realize it multimodally in the given context ("how to express it").

In order to avoid repetitiveness in the long term, it is important to have flexible dialogue strategies and a rich repertoire of verbal and nonverbal expressions to allow for variation. The multimodal input processing module interprets verbal and non-verbal input. Interpretation is guided by information from the user model and the strategic planning (WP2 & 3) and provides information back to them. First, verbal input is needed for the dialog interaction itself as the dialogue's flow takes input from the child to progress. Second, an interpretation of the child's affective state is needed for engagement analysis used in WP2 to adapt the high-level goal self-management goals. Third, feedback is needed for WP3 as basis for adapting a child's preference model, and the long-term memory.

3 Tasks, objectives, results

The overall objective of WP4 is to develop the technologies for personalised multimodal natural interaction serving to actively foster long-term engagement with the robot and its avatar. Voluntary long-term use is required as a prerequisite for other system objectives. This encompasses natural language interpretation and multimodal generation, as well as dialogue management

3.1 Planned work

Our goals during Year 1 have been to establish a baseline for the interaction components. The most important goal was to design a formalism for dialogue specification that builds on knowledge representation to uniformly define and store information about and the structure of natural language and multimodal dialogue. Subsequently, the implementation of components contributing to this formalism had to be started. The existing linguistic resources need adaption to better support natural language generation and interpretation in the PAL context. For the subsequent development and evaluation of the natural language components, interaction data with real users had to be collected.

3.2 Actual work performed

In Year 1, we developed and partly implemented a new methodology for dialogue management and natural language processing which links dialogue processing tightly with knowledge representation in an ontology. The knowledge base serves as a central repository for the specification of atomic semantic elements, such as dialogue acts and action frames, as data storage for an agent's knowledge, it's user model, long-term memory, and belief model. Representing the different chunks of data in a uniform way makes them accessible for interaction management as well as the high-level decision making and child modelling modules.

As a jump start for data collection and first experiments, we also created a prototype out of heavily adapted modules of the predecessor project Aliz-E, with an enhanced version of the verbal generation for Italian to allow for multi-user interactions.

The main work items, which will be detailed out in the next sections, where the following:

- A prototype for experiments and data collection with manageable effort. This prototype will not be sustained and only serves as an interim system until the architecture for the new PAL system is in place
- Definition of a lean formalism to specify dialogue policies that uses a uniform representation of dialogue acts, linguistic and application semantics, and other relevant knowledge that is needed for this task

- The first version of an ontology that specifies how the relevant data is to be represented
- Adapting existing modules and resources, e.g., the HFC reasoner, the existing grammars, etc., for use in PAL
- Collecting interaction data in camp experiments that can be used to build and improve the multimodal interaction modules
- Build a Dutch voice for Mary TTS out of freely available audio resources

T4.1: Robust interpretation of user input

ASR module based on the Google ASR API

We developed a connector to the Google Automatic Speech Recognition (ASR) API that sends its results to the PAL infrastructure. This will serve as an initial component to have speech recognition facilities available in the PAL system, and a baseline to compare other approaches against.

Dialogue act semantics, ontology, knowledge base

For PAL, we pursue a novel approach for dialogue processing, for interpretation as well as for generation of verbal utterances. It is based on a uniform representation of an application semantics that uses dialogue acts and frames that are defined in a RDF/OWL ontology [Hayes, 2004] [ter Horst, 2005]. In addition, all user and other data that influence multimodal generation are specified in the ontology, which facilitates access and combination of the different bits of information. In addition, it provides a full specification of the objects and facts the system can handle, as well from the interaction point of view as from a storage / memory view.

For this purpose, we have extended existing and created new ontologies to accommodate the needs of PAL. One important part contains the dialogue acts specified in the DIT++ standard (see `http://dit.uvt.nl`), and many frames adapted from the FrameNet specification (see `http://framenet.icsi.berkeley.edu/`). Here is a small example for a semantics build from this pieces, and a possible verbalisation:

`Offer(Showing, hasTheme=Picture, sender=I_MYSELF, addressee=NAO_ROBOT, hasTopic=Football)`

I could show you a picture of the last football game.

To make this approach feasible, a software module is necessary that allows to retrieve and add this type of data in a flexible and efficient manner. Here, we use an extended version of the HFC reasoner (annex 4.2.1), build at DFKI, which is comparable in performance with commercial systems.

Furthermore, it contains extensions that are not available in traditional reasoners, e.g., using tuples of more than three elements. This is vital to store information that changes over time, because the time information can be stored directly in the tuple, contrary to standard approaches. Further details about the ontologies and the reasoner are described in the aforementioned annex (4.2.1).

Dialogue Management platform

We defined the first version of a formalism for specifying dialogue actions, in which the atomic specifications, and the information that is used and collected during ongoing dialogues, will be contained in the previously described database. The goal is to provide a dialogue specification language that provides the dialogue designer with the feel of a high-level rule language and introduces convenient shortcuts for the lengthy and complicated database queries that are necessary to retrieve relevant information for the generation and interpretation process. A parser for this language has already been implemented; currently, we are working on the compilation of the parsed representations into low-level programming language constructs and database queries which then constitute the executable dialogue engine.

To arrive at a both usable and principled framework definition, we are using a top-down bottom-up approach. On one hand, we are targeted towards the uniform representation and tight coupling of data representation and dialogue specification. On the other hand, we look at existing prototypes from various projects to verify that our formalism is able to encode these dialogue policies in an efficient and compact way. By approaching this task from an abstract specification, and from concrete examples, we strive for a maximal usability of the resulting system.

T4.2: Flexible and adaptive multimodal generation

Dutch voice for Mary TTS

One goal of the PAL system is to achieve maximal consistency between the virtual and the real version of the PAL companion. To achieve this, using the same text-to-speech (TTS) system would be a major advantage. The NAO robots have a built-in TTS, which unfortunately is not available for free as an independent product.

For Italian, a voice of reasonable quality exists for the free Mary TTS system. This voice has been used successfully in the past, also in conjunction with the NAO robot. Therefore, we pursued the idea of building a free Dutch voice for NAO from existing free audio recordings.

We collected recordings from LibriVox `https://librivox.org/` and the corresponding texts from project Gutenberg `https://www.gutenberg.`

`org/`. For the text-to-phoneme translation, we acquired a licence of the free pronunciation database Fonilex `http://bach.arts.kuleuven.be/fonilex/`

Unfortunately, there were several severe obstacles in voice building process of Mary. Although in the end, we managed to create a unit selection voice from the collected and processed data, its quality was so low that we had to abandon this idea because of the immense amount of work that would still be necessary to improve the result.

Currently, we are looking for alternative possibilities to get Dutch text-to-speech output.

Linguistic Resources (also relevant for T4.1)

Currently, we are using two different approaches for the generation of verbal output. Firstly, we have a template generation engine which is called Content Planner which turns the application semantics, as described above, into a natural language utterance. Every output may consist of a single or more sentences, which can be concatenated or used alternatively. We cover quite a range of dialogue acts and frame arguments with this method already. The advantage is that it is easy to add verbalisations for new inputs, even for non-expert users.

However, planning a natural verbal output using simple strings can become uncomfortable in some cases: e.g. in Italian the adjective inflection is given by the child (or the *activity* name) gender; the type of some prepositions depends on the following word (*conosci la risposta* ***alla prima*** *domanda* vs. ***a questa*** *domanda*). Again, some of the canned text variants affect the word order of the planned utterance. The whole verbalization depends on the interaction type: apart from the different inflection (singular vs. plural), group interactions might require some special utterances which are unwanted/unusual in the single interactions.

Each syntactical variant requires different strings and consequently *different outputs* in the canned text, which makes maintaining the rule consistency more difficult.

On the other hand, *logical forms* (LF), like the semantics in the OpenCCG grammar, represent the words found and sentences using only their canonical form, while attributes like gender or number are provided as feature, not as string. As features can be parameterised more easily than strings, i.e. just using simple variables, substituting the output strings of the canned text with logical forms makes the rule maintenance more comfortable.

In the latest version, the possibility to combine LF and strings was implemented, so that the questions and answers from the Quiz game database can be fully integrated in the LF grammar. Using the OpenCCG surface realizer, we then can build from the defined LF the surface form of the required utterance, which in turn is the input for the TTS system. The work on grammar adaptation is described in detail in annex 4.2.2

Children's perception of the robot as a friend

We analysed data collected during several experiments with diabetic children in Italy, at the San Rafaele hospital in spring 2013 and during summer camps at Misano Adriatico in 2013 and 2014, in order to evaluate how the children's perception of the robot is influenced by their individual interactions with the robot, and by specific relational verbal behaviours. We found that children who interacted with the robot individually perceived it significantly more as a peer and friend than those who only experienced it in non-interactive group sessions. Furthermore, the robot was perceived more as a friend when it displayed familiarity with the child across several sessions and when it elicited the child's self-disclosure in off-activity talk. We also observed increased commitment to interaction success related to familiarity display and increased interest in further interactions related to off-activity talk. These are important findings to incorporate in the design of the interactions and they have consequences for what verbal behaviours need to be modeled, both for generation and for interpretation. Details are presented in annex 4.1.1 and annex 4.1.2.

References

[Hayes, 2004] Hayes, P. (2004). RDF semantics. Technical report, W3C.

[Krieger, 2015] Krieger, H.-U. (2015). Ontologies and reasoning architecture for PAL. Technical report, DFKI GmbH.

[Krieger and Declerck, 2015] Krieger, H.-U. and Declerck, T. (2015). An OWL ontology for biographical knowledge. representing time-dependent factual knowledge. In *Proceedings of the Workshop on Biographical Data in a Digital World*.

[Krieger and Schulz, 2015] Krieger, H.-U. and Schulz, S. (2015). A modal representation of graded medical statements. In *Proceedings of the 20th Conference on Formal Grammar*.

[Krieger and Willms, 2015] Krieger, H.-U. and Willms, C. (2015). Extending OWL ontologies by Cartesian types to represent N-ary relations in natural language. In *Proceedings of the IWCS Workshop on Language and Ontologies*.

[Kruijff-Korbayová et al., 2015a] Kruijff-Korbayová, I., Oleari, E., Bagherzadhal, A., Sacchitelli, F., Kiefer, B., Racioppa, S., and Pozzi, C. (2015a). Young users' perception of a social robot displaying familiarity and eliciting disclosure. In *Proceedings of the 7th International Conference on Social Robotics. International Conference on Social*

Robotics (ICSR-2015), 7th, October 26-30, Paris, France. Springer. to appear.

[Kruijff-Korbayová et al., 2015b] Kruijff-Korbayová, I., Oleari, E., Pozzi, C., Sacchitelli, F., Bagherzadhalimi, A., Bellini, S., Kiefer, B., Racioppa, S., and Sanna, A. (2015b). Let's be friends: Perception of a social robotic companion for children with t1dm. In *Proceedings of the 1st international conference on social robots in therapy and education New Friends 2015. New Friends, October 22-23, Almere, Netherlands*. o.A. to appear.

[Racioppa, 2015] Racioppa, S. (2015). Openccg and lf grammars for PAL. Technical report, DFKI GmbH.

[ter Horst, 2005] ter Horst, H. J. (2005). Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *Journal of Web Semantics*, 3:79–115.

4 Annexes

4.1 Published Peer-Reviewed Papers

4.1.1 Young Users' Perception of a Social Robot Displaying Familiarity and Eliciting Disclosure

Abstract Establishing a positive relationship between a user and a system is considered important or even necessary in applications of social robots or other computational artefacts which require long-term engagement. We discuss several experiments investigating the effects of specific relational verbal behaviors within the broader context of developing a social robot for long-term support of self-management improvement in children with Type 1 diabetes. Our results show that displaying familiarity with a user as well as eliciting the user's self-disclosure in off-activity talk contribute to the user's perception of the social robot as a friend. We also observed increased commitment to interaction success related to familiarity display and increased interest in further interactions related to off-activity talk.

Relation to WP This work studies effects of off-activity talk dialogues during a child-robot session, which provides insides into the dialogue strategies to employ to create engaging dialogues for long-term relationships.

Availability Unrestricted. Included in the public version of this deliverable (annex A.1) [Kruijff-Korbayová et al., 2015a].

4.1.2 Let's Be Friends: Perception of a Social Robotic Companion for children with T1DM

Abstract We describe the social characteristics of a robot developed to support children with Type 1 Diabetes Mellitus (T1DM) in the process of education and care. We evaluated the perception of the robot at a summer camp where diabetic children aged 10-14 experienced the robot in group interactions. Children in the intervention condition additionally interacted with it also individually, in one-to-one sessions featuring several game-like activities. These children perceived the robot significantly more as a friend than those in the control group. They also readily engaged with it in dialogues about their habits related to healthy lifestyle as well as personal experiences concerning diabetes. This indicates that the one-on-one interactions added a special quality to the relationship of the children with the robot.

Relation to WP The perception of the robot is an important point in building a lasting relationship. Verbal interaction poses multiple opportunities to shape the right perception, and make the robot a trusty companion.

Availability Unrestricted. Included in the public version of this deliverable (annex A.2) [Kruijff-Korbayová et al., 2015b].

4.1.3 An OWL Ontology for Biographical Knowledge. Representing Time-Dependent Factual Knowledge

Abstract Representing time-dependent information has become increasingly important for reasoning and querying services defined on top of RDF and OWL. In particular, addressing this task properly is vital for practical applications such as modern biographical information systems, but also for the Semantic Web / Web 2.0 / Social Web in general. Extending binary relation instances with temporal information often translates into a massive proliferation of useless container objects when trying to keep the underlying RDF model. In this paper, we argue for directly extending RDF triples with further arguments in order to easily represent time-dependent factual knowledge and to allow for practical forms of reasoning. We also report on a freely available lightweight OWL ontology for representing biographical knowledge that models entities of interest via a tri-partite structure of the pairwise disjoint classes *Abstract*, *Object*, and *Happening*. Even though the ontology was manually developed utilizing the Protégé ontology editor, and thus sticking to the triple model of RDF, the meta-modelling facilities allowed us to cross-classify all properties as being either synchronic or diachronic. When viewing the temporal arguments as "extra" argument

Relation to WP To implement a dialogue or long-term memory, but also for many other aspects of dialogue management, finding an efficient representation of information that changes over time is essential. Insofar these results have direct applications in the treatment of dialogue phenomena.

Availability Unrestricted. Included in the public version of this deliverable (annex A.3) [Krieger and Declerck, 2015].

4.1.4 A Modal Representation of Graded Medical Statements

Abstract Medical natural language statements uttered by physicians are usually *graded*, i.e., are associated with a degree of uncertainty about the validity of a medical assessment. This uncertainty is often expressed through specific verbs, adverbs, or adjectives in natural language. In this paper, we look into a representation of such graded statements by presenting a simple non-standard *modal logic* which comes with a set of modal operators, directly associated with the words indicating the uncertainty and interpreted through confidence intervals in the model theory. We complement the model by a set of RDFS-/OWL 2 RL-like entailment (if-then) rules, acting on the syntactic representation of modalized statements. Our interest in such a formalization is related to the use of OWL as the de facto standard in (medical) ontologies today and its weakness to represent and reason about assertional knowledge that is uncertain or that changes over time. The approach is not restricted to medical statements, but is applicable to other graded statements as well.

Relation to WP One important source of information for the PAL system consist in medical assessments in written text. To analyse these texts and make the results readily available in an abstract, but adequate form may greatly support the decision making modules in PAL.

Availability Unrestricted. Included in the public version of this deliverable (annex A.5) [Krieger and Schulz, 2015].

4.1.5 Extending OWL Ontologies by Cartesian Types to Represent N-ary Relations in Natural Language

Abstract Arbitrary n-ary relations ($n \geq 1$) can in principle be realized through binary relations obtained by a reification process that introduces new individuals to which the additional arguments are linked via accessor properties. Modern ontologies which employ standards such as RDF and OWL have mostly obeyed this restriction, but have struggled with it nevertheless. Additional arguments for representing, e.g., valid time, grading, uncertainty, negation, trust, sentiment, or additional verb roles (for ditransitive verbs and adjuncts) are often better modeled in relation and information extraction systems as direct arguments of the relation instance, instead of being hidden in deep structures. In order to address non-binary relations directly, ontologies must be extended by Cartesian types, ultimately leading to an extension of the standard entailment rules for RDFS and OWL. In order to support ontology construction, ontology editors such as Protégé have to be adapted as well.

Relation to WP This work describes extensions of software modules to handle temporal representations such as those in [Krieger and Declerck, 2015] more efficiently. This directly supports the technical basis for WP4.

Availability Unrestricted. Included in the public version of this deliverable (annex A.4) [Krieger and Willms, 2015].

4.2 Technical Reports

4.2.1 Hans-Ulrich Krieger (2015), "Technical Report: Ontologies and Reasoning Architecture for PAL"

Abstract This technical report contains the work on ontologies for dialogue processing and user modeling in PAL, and methodological and technical adaptations to an existing implementation of a reasoner for extensions of Description Logic.

Relation to WP The reasoner and ontologies are the foundation of the verbal communication platform that we intend to develop in PAL. The flexible and uniform representation it provides for all relevant kinds of data makes it a central point in work package 4, and contributes to all tasks in this work package.

Availability Unrestricted. Included in the public version of this deliverable (annex B.2).

4.2.2 Stefania Racioppa (2015), "OpenCCG and LF Grammars for PAL"

Abstract In the past year, we developed a large Italian OpenCCG grammar covering all currently used utterances in Italian end English. In the current version, this grammar generates for the parsed utterances the same semantic output as the English OpenCCG grammar Moloko, which was used in several DFKI projects like CogX and NIFTi. For Italian, we also developed a LF (logical form) transfer grammar, integrating the semantic output of the Italian OpenCCG grammar in the Content planner: this version supports all the Speech acts and variants defined in the current template generation version, but in the output side the strings and string parts were substituted by the logical forms parsed by the OpenCCG grammar. This makes the grammar maintenance and the realizer parameterization more comfortable and flexible. The string input for the TTS module is then generated by the OpenCCG surface realizer. Besides this, the OpenCCG grammar was improved in terms of coverage and parsing precision, the verbalization for the interaction with groups was implemented in the Content planner grammars, and the questions and answers strings from the Quiz game database were fully integrated in the LF grammar.

Relation to WP Directly relates to task T4.1.

Availability Unrestricted. Included in the public version of this deliverable (annex B.1).

A Publicly available papers

Young Users' Perception of a Social Robot Displaying Familiarity and Eliciting Disclosure

Ivana Kruijff-Korbayová[1], Elettra Oleari[2], Anahita Bagherzadhalimi[2], Francesca Sacchitelli[2], Bernd Kiefer[1], Stefania Racioppa[1], Clara Pozzi[2], Alberto Sanna[2]

Language Technology Lab, DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany,
{ivana.kruiff,bernd.kiefer,stefania.racioppa}@dfki.de,
Fondazione Centro San Raffaele (FCSR), vial Olgettina 60, 20132 Milan, Italy,
{oleari.elettra,bagherzadhal.anahita,sacchitelli.francesca,pozzi.clara,sanna.alberto}@hsr.it

Abstract. Establishing a positive relationship between a user and a system is considered important or even necessary in applications of social robots or other computational artifacts which require long-term engagement. We discuss several experiments investigating the effects of specific relational verbal behaviors within the broader context of developing a social robot for long-term support of self-management improvement in children with Type 1 diabetes. Our results show that displaying familiarity with a user as well as eliciting the user's self-disclosure in off-activity talk contribute to the user's perception of the social robot as a friend. We also observed increased commitment to interaction success related to familiarity display and increased interest in further interactions related to off-activity talk.

Keywords: child-robot interaction; human-robot interaction; long-term interaction; social robot; verbal behavior; personalization; continuity behaviors; familiarity display; self-disclosure; off-activity talk; perception of robot as friend

1 Introduction

It has become a commonplace vision that robots will partake in many areas of our lives. The role they are envisaged to fulfill has shifted from that of a mere tool to a teammate, peer, companion, friend. Thus, being conceived of as social actors, which will be explicitly and intentionally entering into relationships with humans. Social science research has identified a plethora of behaviors that are prevalent and influential in establishing and maintaining human-human relationships. Inspired by the seminal work on relational agents by Bickmore and colleagues [3] a growing body of research now studies what effects do such behaviors have in human-machine, and more specifically human-robot relationships, and how we can implement the corresponding functionality to enable machines/robots to perform these behaviors autonomously. Overviewing this body of literature, it is clear that the more we know, the more we know what we do not know. There remain many aspects to be studied.

adfa, p. 1, 2011.

The research presented here concerns relational verbal behaviors that contribute to the perception of an agent as a friend. It is set within the broader vision of developing a robotic companion to provide long-term support to children with Type 1 Diabetes Mellitus (T1DM) to help them learn and improve their ability to independently manage their condition. During the process of self-management development, children with T1DM need to acquire knowledge about diabetes and suitable healthy nutrition, develop various relevant skills and learn to adhere to the therapy requirements. Similarly to what was noted for health behavior change applications [3], establishing and maintaining a positive relationship is considered to be a necessary (though likely not sufficient) condition for addressing the further goal of influencing diabetes self-management. In this paper we focus on two aspects of relational verbal behavior which personalize an interaction by linking it to the experiences of a given user: signaling continuity over time by references to joint experiences of the user and the robot in interaction with one another (Familiarity Display – FD); and eliciting disclosure about separate experiences of the user (Off-Activity Talk - OAT). In a series of experiments with an implemented integrated system, comparing independently a condition with and without FD and with and without OAT, we found that these behaviors contribute to young users' perception of the robot as a friend. We first review related work on such relational verbal behaviors in Sec. 2. In Sec. 3 we describe our system. In Sec. 4 and 5 we present the methodology and results of the experiments addressing FD and OAT, respectively. We discuss the observed effects and conclude in Sec. 6.

2 Background

Bickmore and colleagues developed the concept of relational agents, referring to computational artifacts designed to establish and maintain long-term social-emotional relationships with their users [3]. They discussed a myriad of strategic relational behaviors, instantiated them in systems and carried out numerous studies to evaluate the effects of various aspects of relational agent behavior on long-term engagement and behavior change, e.g., [4]. This inspired many other researchers to perform further studies and experiments in this area. What we call familiarity display has been called continuity behaviors in some previous literature. For example, the continuity behaviors implemented in the FitTrack system [3] and the person memory model of a virtual agent described in [13] include greetings and farewells referring to past/future encounters and reference to mutual knowledge, e.g., user's biographical facts, preferences and interests mentioned in a previous session. The exercise advice system described in [8] also implements continuity behaviors as means of relationship maintenance, namely reference to previously given advice and gradually more personal greetings, including some small talk. Various other systems have included a user model or some form of long-term memory and used it to refer to content from previous interactions [1, 5, 16, 18].

Our concept of Off-Activity Talk corresponds to the reciprocal self-disclosure discussed as another relational behavior and found to increase trust, closeness and liking in work cited by [3]. While the OAT in our system allows reciprocity, we have focused on eliciting disclosure from the users so far. This resembles the gathering of personal information in [1, 5, 13, 18], but is more conversational.

A comparison of existing results concerning the effects of various relational behaviors is complicated by the fact that each study uses measures and methodologies adjusted to its purpose. For example, [3] evaluated the effects of all the relational behaviors combined. They found an effect on long-term relationship, but not on behavior change in a real usage longitudinal study. On the other hand, [8] evaluated the isolated impact of relationship maintenance on users' attitudes and found an effect on various metrics and [13] investigated the impact on social presence, likability and communication satisfaction of using personal information during the interaction sessions. These studies were done with adults, the systems of [3] and [8] were not robots, and the metrics did not include a classification of the user's perception of their relationship with the system. Some of the experiments did not involve usage in real life, but the participants used the system to play out hypothetical situations, e.g. [8].

3 System and Setup

Our experiments were carried out with the system developed gradually in the course of the Aliz-E project [7]. The robot we use is the small humanoid robot Nao from Aldebaran Robotics. The system integrates components for speech recognition and interpretation as well as natural language generation and synthesis, gesture capture and interpretation, nonverbal behavior production and motor control, activity-, interaction- and dialog management, and a user model to store key information about each child [9]. Several game-like activities were implemented in the system: Quiz, Imitation, Dance and Collaborative Sorting [9, 2, 15]. A range of relational social behaviors reported in the literature was implemented across the activities, including informal greetings, introductory small talk, the use of first names, empathy related to the performance in an activity, the robot's ability to make mistakes, nonverbal bodily cues, allowing children to touch the robot [14]. Although the robot was presented as autonomous to the participants, we relied on a partially Wizard-of-Oz setup, where a human Wizard simulated the speech and gesture input interpretation, could override the automatic dialog management decisions, if needed, and fully controlled off-activity talk.

4 Experimental Study 1: Familiarity Display

The first study was a longitudinal experiment investigating the use and effects of continuation behaviors. We investigated how the robot can acquire familiarity with a user and display it in interactions, and what effect this would have on children's perception of the robot.

4.1 Familiarity Display

When humans interact with each other over a series of encounters, they become familiar, i.e., they accumulate shared knowledge (shared history, personal common ground) [6]. The goal of this study was to endow the robot with the ability to acquire a persistent interaction history respective to each individual user and allow it to manifest its familiarity with the user both verbally and nonverbally later in the same interaction or in subsequent interactions. We selected several parameters that the robot would use to represent the interaction history: the user's name; whether it is the first or a subsequent encounter of the user with the robot; for each activity whether the user has already performed it or not and some details about it (e.g.: for each Quiz question, whether it has been asked before in a interaction with the user); the user's last performance on each activity. The values of these parameters for each user were stored in a persistent user model. We designed templates for verbal output generation which allowed to include content based on the user model. The robot would use these verbalizations to explicitly display its familiarity with the user. Such verbal moves would also be accompanied by nonverbal behaviors showing familiarity, e.g., nodding, higher excitement. We also designed alternative verbalizations which were neutral, i.e., they would not show whether the robot is or is not familiar with the user. Examples of verbalizations of both kinds are shown in **Table 1**.

Table 1. Examples of verbalizations that signal familiarity (used in the FD condition, see paragraph 4.2) or are neutral in this respect (used in the ND condition, see paragraph 4.2).

Familiarity display	**Neutral display**
Use of user's name: So, which answer do you choose, *Marco*?	So, which answer do you choose?
References to previous encounters and play experiences: I am happy to see you *again*. It was nice playing with you *last time*.	I am happy to see you. -
References to previous performance in an activity: Are you ready to play *again*? Today you were *again* very good at quiz. Well done, you've done *better than last time*.	Are you ready to play the quiz? Today you were really good at quiz. Well done.

4.2 Experiment Methodology

As described in detail in [10], 19 children participated in total (11 male, 8 female; age 5-12, all Italian), of which 13 participated in three sessions on different days as fore-seen in the protocol (9 male, 4 female; 6 with T1DM, 7 healthy).

We exerted a between-subjects design with two conditions: the Familiarity-Display (*FD* - 9 children) condition and the Neutral Display (*ND* - 10 children) condition. The robot used the verbal and nonverbal behaviors described in Sec. 4.1, respectively.

The experiment took place at a research lab at the San Raffaele hospital in Milan. The sessions were organized on several Saturdays over a period of two months and full participation involved three sessions on different dates per child, where s/he could choose among one (or more, time permitting) of the available activities to be performed with the robot: Quiz, Imitation game and Dance. Each session of the experiment lasted maximally one hour, including the interaction session with the robot and filling in 3 post-interaction questionnaires. These latter were multiple choice questionnaires reporting the child's self-assessment of: (Q1) *the perceived bond with the robot*, to be categorized between different levels of confidence and familiarity: stranger, neighbor, classmate, teacher, friend, relative, sibling, parents; (Q2) *the perceived role during the activities*: child leading, robot leading or on a par; (Q3) *the perception of the robot*: through a multi-adjective choice among friend, toy, pet or game console. Children were also asked to briefly explain their choices. The questionnaires were administered to the participants at the end of each session, in order to see if there was any change over time.

4.3 Results

We analyzed the post-interaction questionnaires, linking to each multiple choice answer a numerical value. We calculated the means and standard deviations of the scores per child across the interaction sessions.

Questionnaires Q1 and Q2 did not reveal any statistical significance regarding the perception of either the bond with the robot or the level of the established relationship, neither between the two experimental conditions (*FD* and *ND*) or across the sessions (for those children who interacted three times). From the explanations that the participants gave to justify their answers, as a qualitative insight we saw that, independently from the two conditions, high rates of perception of the bond (from friend to parent) were related to the play dimensions (e.g.: "having fun" and "play together") and the friendly approach ("it's nice/cute/tender") that the robot showed to children. Lower values, linked to the perception of different levels of relationship (stranger, neighbor, teacher), were mainly related to a low satisfaction and engagement in the activity/ies performed (e.g. "too difficult questions/tasks", "questions like homework", etc.). In addition, there was an overall perception of the interactions with the robot as being "at the same level".

An interesting result was found in Q3: a comparison of the adjective choices revealed that all 9/9 children in the *FD* condition perceived the robot as friend after the first session as opposed to the only 4/10 *ND* children (Fisher's test: two-tailed P=0.0108). Among the 13 children who continued to have 3 interactions 5/6 *FD* children maintained the perception of the robot as friend, only one changed it to a toy. No trend was observed among the 7 *ND* children.

5 Experimental Study 2: Off-Activity Talk

The second study investigated the effects of Off-Activity Talk (OAT) in one-on-one interaction sessions held in the context of two different educational summer camps for children with T1DM, organized by the Italian patient's association SOStegno70 (www.sostegno70.org) and the pediatric department of San Raffaele hospital

(Milan). We investigated how the robot can engage a child in OAT on topics related to diabetes and healthy lifestyle and how this would impact children's perception of: their relationship to the robot; interest to undergo further interactions with it; willingness to talk about diabetes and self-disclosure; adherence to a therapy-related requirement to fill in a nutritional diary. The methodology and results of the 2013 experiment have been published in [11, 12]. We only repeat those points that are relevant for a comparison with the 2014 experiment.

5.1 Off-Activity Talk

OAT are sub-dialogs interspersed within the talk pertaining to a particular activity, without being directly part of it [11, 12]. The main OAT topics we defined were pertinent to: hobbies, diabetes, nutritional habits, relationships with friends and adults with respect to diabetes (for examples, see [11]). For each topic, several OAT-initiation system utterances were formulated and implemented as canned text in the system. The initiation of OAT was triggered by the Wizard at opportune moments during the interaction(s) within the course of an activity. For example, during the Collaborative Sorting game, OAT about nutritional habits was triggered while the child was playing a virtual game about the food and carbohydrates counting (see the example extracted from a longer interaction in the left column in **Table 2**).

Table 2. Examples of OAT subdialogs from the 2014 experiment (translated from Italian). *Left*: OAT on nutritional habits linked to the contents of the Sorting Game. *Right:* Child self-disclosure concerning diabetes.

Nao: *what is your favourite food among these?* Child: *among the remaining ones - ice cream. I really like it. although I can't always eat it, it's not good for me.* Nao: *which flavour do you prefer?* *Child: <EH> stracciatella.* Nao: *that's good.* Child: <*laughter*> Nao: *can you distinguish foods according to carbohydrates?* Child: *yes, I do count - as I do for the insulin at home. I count carbohydrates.*	Nao: *do you measure glycaemia yourself at home?* Child: *yes, I do it myself, because now I'm quite big, so -* Nao: *you're so good.* Child: *thanks.* Nao: *I was told that it is difficult.* Child: *<EH> a - well, it is a bit difficult at the beginning, because even - say - accept that you have diabetes is a bit difficult. but then you get used to it. […]* Nao: *I'd like to ask you how do you do it.* Child: *how do I measure glycemia?* Nao: *yes.* Child: *so, basically <EH> I take my measuring set, I disinfect the finger I want to prick, I take the pen, prick and then put the blood on the strip, which automatically measures glycaemia, so that I see the value on the screen of the set.*

5.2 Experiment Methodology and Results

Participants. The experiments for this study have been conducted in two different educational summer camps in 2013 and 2014. In the 2013 experiment, 20 children (age 11–14, 10 females and 10 males) among the total of 59 attending the camp volunteered to participate in the individual sessions with the robot. In 2014 it was 28[1] (age 10–14, 10 females and 18 males) out of 41. The remaining children were in both cases included in the control group and experienced the robot in the camp only as a theater performance character during recreational evening activities.

Procedure. In both the 2013 and 2014 summer camps, children who volunteered for individual session(s) with the robot were given an appointment during their spare time at the camp. Before beginning the interaction, they were instructed about the available game activities with the system and the possibility to freely choose among them during their session. The session lasted a maximum of 30 minutes. The interactions were carried out using the system described in Sec. 3.

2013 Camp Experiment overview. The specific objectives of the 2013 camp were to investigate the feasibility and acceptance of OAT, its effects on children's perception of the robot and on adherence to medical advice (i.e.: filling in a nutritional diary). The study was carried out in a between-subjects design with 3 conditions: (1) *OAT*: one-on-one interaction with the OAT feature turned on; (2) *non-OAT*: one-on-one interaction without OAT; (3) *CONTROL*: no one-on-one interaction.

The results related to this study are discussed in detail in [11, 12] but with respect to the present contribution, it is interesting to mention that qualitatively children's acceptance of OAT was good: they engaged in it readily and elicited self-disclosure from the robot [12]. However, their responses to the robot's OAT prompts were brief and concise, maybe due to their formulation as closed questions. Moreover, the presence of OAT turned out to have a positive effect on the children's interest to interact with the robot again: although all subjects in the two intervention conditions expressed interest to play again with the robot, only 11 actually booked another slot: 9/10 in the *OAT* group and 2/10 in *non-OAT* (Fisher's test, two-tailed P=0.0055).

2014 Camp Design. Based on the positive experience with OAT in the 2013 experiment, we decided to drop the non-OAT condition. The 2014 experiment thus had a between-subject design with the *OAT* and the *CONTROL* condition. We revised the OAT prompts, to include more open questions or clusters of closed interconnected questions, in order to elicit more complex OAT dialogs with more child talk. **Table 3** shows some examples of these variations; **Table 2** shows OAT interaction examples.

Table 3. Examples of the different verbalization of the OAT prompts used in the two Camps.

2013 OAT prompts formulation	2014 OAT prompts formulation
Can you draw?	Can you draw? What do you like to draw?
Do you realize when your glycaemia is low?	Do you realize when your glycaemia is low? What do you do in these cases?
What is the strangest food you've ever tried?	What is the strangest food you've ever tried? Where were you when you tried it? Abroad?

[1] The data of one subject was discarded as the child did not finish the interaction.

We also further elaborated the evaluation of children's perception of the robot. We designed a new questionnaire composed of two closed questions. The first one asked to describe the robot by choosing one out of the following set words: friend, toy, pet, adult, computer. The second one asked to choose one of 16 listed adjectives describing the robot. The adjectives belonged to three categories of perception: machine (e.g. fake, scientific, etc.), relational (e.g. interested in me, someone to trust, etc.), humanized (e.g. spontaneous, empathetic, etc.). This questionnaire was administered to all the participants of the camp at the end of their stay. Furthermore, to evaluate children's willingness and spontaneity to talk about diabetes, we performed an analysis of the interactions similar to the one described in [12]: 3 coders (native speakers) evaluated every OAT sub-dialog regarding diabetes on a 4 point scale (i.e.: 1= "not responding or not willing at all", 2= "forced or annoyed", 3="clear, simple and courteous", 4="very interested and active") as well as assigned an overall score per child to how the OAT diabetes sub-dialog were going.

2014 Camp Results. OAT had an effect on the children's perception of their relation-ship to the robot: 26/27 in the *OAT* group and only 4/13 in the *CONTROL* group selected the word "friend" among the 5 options offered in the questionnaire. The difference between the two proportions is strongly statistically significant (χ^2=20.09 with probability 1%, two-tailed p=0.0001). Regarding the multiple adjective choice, even if not supported by statistical significance, we observe that children in the *OAT* condition chose no machine category adjectives, 30% of the chosen adjectives belonged to the humanized category and 70% to the relational one. Whereas in the *CONTROL* condition 20% of the adjectives chosen belonged to the machine category, 20% to the humanized one and 60% to the relational one. The children's willingness and spontaneity to engage in OAT and talk about diabetes was high. Moreover, the coders noticed qualitatively a common attitude of the children in sharing their practical notions about diabetes with the robot and their personal experiences on what it is like to deal with diabetes in their daily lives (see the excerpt in the right column of **Table 2**).

6 Discussion and Conclusion

We described a series of experiments with a robotic multi-activity system designed to provide long-term support to children with T1DM. We addressed the potentialities of specific relational verbal behaviors in contributing to the perception of a robotic character as a friend by the young participants: familiarity display and off-activity talk. Both these features were introduced in order to personalize the interactions in a way that resembles typical human interactions between friends: making reference to joint experiences and fostering self-disclosure about personal topics (in this case diabetes- and health related topics). We found that children interacting with the robot displaying familiarity, clearly perceived it as a friend after the first interaction as well as after three interactions in a longitudinal study. They also felt to have been at the same level of control with the robot during the interactions. This outcome was also confirmed by the investigations of the 2014 summer camp experiment, carried out with a different set of children in a real world setting, even though the set of words available to define the role of the robot was slightly different on the two occasions. In the 2014 summer camp experiment the set of choices included also the word “adult” in order to allow for a difference in the level of the perceived relationship biased towards the robot (robot compared to a figure that usually leads situations), rather than towards the child (as in the case of a pet or a video game). Confirming the previous results, none of the children chose this description. As for Off-Activity Talk, children were at ease during the dialogs with the robot and seemed to appreciate the interest that it showed for their daily lives and experiences. The combination of these factors led to a natural adaptation of children’s behaviors to the specific single interaction dynamics and triggering, a spontaneous conversation regarding the delicate topic of diabetes. Moreover, the dialog structure enriched with the OAT prompts seemed to be a key factor in engaging children and making them interested to interact again with the system. This is a significant achievement in the long term perspective of our research, even though more longitudinal studies are needed to address this point. To conclude, the fact that the robot is perceived by children as a friend capable to establish and maintain a positive relationship is extremely impactful in a broader real life application perspective of a robotic companion. Children could be more inclined to feel at ease and open themselves with such a robot, thus offering the diabetology teams of caregivers a valuable instrument to support their work of education, addressing the goal to improve self-management of young patients.

Acknowledgments

The research presented in this paper has been funded by the Aliz-E project (alize.org), grant n° ICT-248116 in FP7/2007-2013, and the PAL project (pal4u.eu), grant n° 643783-RIA in Horizon 2020. We wish to thank the Sostegno70 association for diabetic children and the team of the Paediatric Unit of Ospedale San Raffaele for their constant support in this research.

References

1. Adam, C., Cavedon, L., and Padgham, L. Hello emily, how are you today? Personalised dialogue in a toy to engage children. In Proceedings of the 2010 Workshop on Companionable Dialogue Systems, Association for Computational Linguistics (ACL), pp. 1–6, Uppsala, Sweden (2010).
2. Baxter P., Wood R., and Belpaeme T. A Touch screen-Based Sandtray to Facilitate, Mediate and Contextualize Human-Robot Social Interaction. In Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI, pp. 105-106, Boston, MA, U.S.A (March 2012).
3. Bickmore, T.W. and Picard, R.W., Establishing and maintaining long-term human-computer relationships. ACM Transactions on Computer-Human Interaction (TOCHI), vol. 12(2), pp. 293 – 327 (June 2005).

4. Bickmore, T., Schulman, D., and Yin, L., Maintaining engagement in long-term interventions with relational agents, International Journal of Applied Artificial Intelligence, special issue on Intelligent Virtual Agents, vol. 24(6), pp. 648–666 (2010).
5. Campos, J. C. F., May: My memories Are Yours. An interactive companion that saves the users memories. Master thesis, Instituto Superior Técnico (2010).
6. Clark, H., Using Language. Cambridge University Press, Cambridge, (1996).
7. Coninx, A., Baxter, P., et al. (in press). Towards Long-Term Social Child-Robot Interaction: Using Multi-Activity Switching to Engage Young Users. To appear in *Journal of Human-Robot Interaction*. http://www.coninx.org/work/ALIZE-JHRI-preprint.pdf
8. De Boni, M., Richardson, A., and Robert., H., Humour, relationship maintenance and personality matching in automated dialogue: A controlled study. Interacting with Computers, 20(3), pp.342–353 (May 2008).
9. Kruijff-Korbayová, I., Athanasopoulos, G., Beck, A., Cosi, P., Cuayahuitl, H., Dekens, T. et al (2011). An Event-Based Conversational System for the Nao Robot. In IWSDS2011 Workshop on Paralinguistic Information and its Integration in Spoken Dialogue Systems, pp. 125-132. Springer (2012).
10. Kruijff-Korbayová, I., Baroni, I., Nalin, M., Cuayahuitl, H., Kiefer, B., Sanna, A., Children's Turn-Taking Behavior Adaptation in Multi-Session Interactions with a Humanoid Robot in: Guy Hoffman, Maya Cakmak, Crystal Chao (eds.): Workshop on Timing in Human-Robot Interaction, Bielefeld, Germany, ACM/IEEE, ACM/IEEE, (2014).
11. Kruijff-Korbayová, I., Oleari, E., Baroni, I., Kiefer, B., Zelati, M. C., Pozzi, C. et al (2014). Effects of Off-Activity Talk in Human-Robot Interaction with Diabetic Children. In Ro-Man 2014: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, pp. 649 – 654, Edinburgh, United Kingdom, (2014).
12. Kruijff-Korbayová, I., Oleari, E., Pozzi, C., Racioppa, S., Kiefer, B., Analysis of the Responses to System-Initiated Off-Activity Talk in Human-Robot Interaction with Diabetic Children. Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue, pp. 90-97, Edinburgh, United Kingdom, (2014).
13. Mattar, N., Wachsmuth, I., Let's Get Personal - Assessing the Impact of personal Information in Human-Agent Conversation. In HCI (2), (2014).
14. Nalin, M., Baroni, I., Sanna, A., Pozzi, C., Robotic Companion for Diabetic Children. In Conference Interaction Design for Children, pp. 260-263, (2012).
15. Ros, R., Baroni, I., Demiris, Y., Adaptive Human Robot Interaction in Sensorimotor Task Instruction: From Human to Robot Dance Tutors. Robotics and Autonomous Systems, Vol 62(6, V), pp. 707-720, (2014).
16. Schröder, M., and Trouvain, J., The German text-to speech synthesis system MARY: A tool for research, development and teaching. International Journal of Speech Technology, 6(4), pp. 365–377 (2003).
17. Schulman, D., Bickmore, T., and Sidner, C. L., An intelligent conversational agent for promoting long-term health behavior change using motivational interviewing. In Proceedings of AAAI (2011).
18. Sieber, G., and Krenn, B., Episodic memory for companion dialogue. In Proceedings of the 2010 Workshop on Companionable Dialogue Systems, Association for Computational Linguistics (ACL), pp. 1–6, Uppsala, Sweden (2010).

Let's Be Friends: Perception of a Social Robotic Companion for children with T1DM

Ivana Kruijff-Korbayová[a], Elettra Oleari[b], Clara Pozzi[b], Francesca Sacchitelli[b], Anahita Bagherzadhalimi[b], Sara Bellini[b], Bernd Kiefer[a], Stefania Racioppa[a], Alexandre Coninx[c], Paul Baxter[d], Bert Bierman[e], Olivier Blanson Henkemans[e], Mark Neerincx[e], Rosemarijn Loije[e], Yiannis Demiris[c], Raquel Ros Espinoza[c], Marco Mosconi[b], Piero Cosi[f], Rémi Humbert[g], Lola Cañamero[h], Hichem Sahli[i], Joachim de Greeff[d], James Kennedy[d], Robin Read[d], Matthew Lewis[h], Antoine Hiolle[h], Giulio Paci[f], Giacomo Sommavilla[f], Fabio Tesser[f], Georgios Athanasopoulos[i], Georgios Patsis[i], Werner Verhelst[i], Alberto Sanna[b], Tony Belpaeme[d]

[a]*DFKI, Language Technology Lab (Saarbruecken, Germany)*
[b]*Fondazione Centro San Raffaele, eServices for Life and Health (Milano, Italy)*
[c]*Imperial College London (London, United Kingdom)*
[d]*The Cognition Institute, Plymouth University (Plymouth, United Kingdom)*
[e]*Organization for Applied Scientific Research (Soesterbergs, The Netherland)*
[f]*National Research Council – ISTC (Padova, Italy)*
[g]*Aldebaran Robotics (Paris, France)*
[h]*University of Hertfordshire (Hatfield, United Kingdom)*
[i]*Vrije Universiteit, Electronics & Informatics Dept. (Brussel, Belgium)*

Abstract. We describe the social characteristics of a robot developed to support children with Type 1 Diabetes Mellitus (T1DM) in the process of education and care. We evaluated the perception of the robot at a summer camp where diabetic children aged 10-14 experienced the robot in group interactions. Children in the intervention condition additionally interacted with it also individually, in one-to-one sessions featuring several game-like activities. These children perceived the robot significantly more as a friend than those in the control group. They also readily engaged with it in dialogues about their habits related to healthy lifestyle as well as personal experiences concerning diabetes. This indicates that the one-on-one interactions added a special quality to the relationship of the children with the robot.

Keywords: Social robots, Child-Robot Interaction, diabetes, Off-Activity Talk, self-disclosure, social skills, social robot perception.

INTRODUCTION

Type 1 Diabetes Mellitus (T1DM) is a chronic disease that affects a shocking 17,000 new children, mostly under 14 years old, per year in Europe [1]. T1DM is an overwhelming pathology that can cause life-threatening complications. It requires children of all ages to learn to constantly manage their condition in terms of glycaemia monitoring and insulin injection. This necessitates a major change in their lifestyle [2].

The present work stems from the Aliz-E project [3], in which we investigated the use of a humanoid social robot to support children with T1DM on their way to self-management. A social robot system was developed and instantiated in a Robot Theatre to facilitate child-robot interaction [4]. It was deployed in real-life settings during two editions of a Diabetes Summer Camp in 2013 and 2014, organized by the Italian families association "Sostegno70 – insieme ai ragazzi diabetici ONLUS" and the team of the Pediatric unit of Ospedale San Raffaele (Milan, Italy).

During the 2013 summer camp we experimented with introducing so-called *Off-Activity Talk* (OAT) to engage children in conversations about topics related to diabetes and healthy lifestyle as part of one-on-one interactions around gaming touchpoints with the robot. Details about the experiment design and a comparison of the effects of individual interactions with and without OAT were presented in [5]. We also observed that children who participated in the individual interactions exhibited a significantly stronger adherence in following the medical advice to fill in a nutritional diary than children who only participated in group interactions with the robot.

We hypothesized that this might be due to a different quality of the child-robot relationship established through the individual interaction. This inspired us to further investigate the effect(s) of individual interactions on children's perception of the robot during the 2014 edition of the camp. This paper presents the method and the results of the 2014 experiment.

EXPERIMENT GOALS AND METHODOLOGY

Goals

The aim of the 2014 summer camp experiment was to further investigate the children's (i) perception of the social robotic companion; (ii) expectations about having a robotic companion in their daily life; (iii) willingness and spontaneity to talk freely about their diabetes condition.

Design

The experiment was held in August 2014 during a ten-day-long Diabetes Summer Camp for T1DM children. All the children at the camp had the opportunity to experience the robot in scripted "theater" performances during collective evening recreational activities. Out of the 41 children attending the camp, 28 volunteered to participate in the study.

The study had a between-subject design with two conditions: (1) the *control* condition, constituted by children who only experienced the social robot as a theater-performance character, but did not interact individually with it; (2) the *intervention* condition, where children had the additional possibility to interact individually with the social robot.

The individual interactions for the *intervention* condition were carried out using the Robot Theatre described in [4] in a partially Wizard-of-Oz setup and were centered around three activities, among which the children could freely choose and switch: a quiz game, a sorting game and a creative dance activity (see Figure 1). More details about the activities can be found in [4] and [5].

During these interactions the robot elicited off-activity-talk as described in [5] and exhibited the following social behavior characteristics discussed in [6]: the ability to express recognition and familiarity (e.g., using the child's name, referring to previous joint experiences); non-verbal bodily cues [7]; turn taking during game playing [8][9]; allowing children to touch it and responding to touch; and occasionally making mistakes, which helps children to feel at ease.

Measures

Children's perception of the robot and their expectations about the possibility to have a robotic companion were measured by questionnaires. Children's willingness and spontaneity to talk about diabetes was evaluated by 3 raters who independently assessed every OAT sub-dialogue regarding diabetes.

Figure 1: Left-to-right:
the quiz game, the sorting game, the creative dance activity

RESULTS

The robot was described as a friend (as opposed to pet, toy, adult, computer) significantly more often by the intervention group than the control (χ^2=20.09 with probability 1%, two-tailed p=0.0001). Instead, there was a tendency in the control group to ascribe machine-like characteristics to the robot, unlike in the intervention group. The children's willingness and spontaneity to talk about diabetes was mostly high. Qualitatively, all coders noticed a common positive attitude in sharing practical notions about diabetes and often also their personal experiences with the robot. Majority of children in the intervention group would like to meet the social robotic companion again (more preferred at home rather than school, hospital, or summer camp) or own one. The reason was the playful character or the relational aspect in majority of cases. This unique relationship also had a positive impact on the educational aspects of the interaction.

CONCLUSIONS

The individual interactions lead the children to perceive the robot as a peer. They do not feel judged, but rather encouraged to learn and exchange knowledge. This finding underlines the potential of such a robotic companion. It shows that children are willing to let a robot enter such a delicate and personal dimension. This is extremely important for fostering companionship to support children with diabetes.

REFERENCES

[1] http://www.idf.org/diabetesatlas/europe

[2] Freeborn D., Dyches T., Roper S.O., Mandleco B. (2103) Identifying challenges of living with type 1 diabetes: child and youth perspectives. *Journal of Clinical Nursing*. 22/13-14. 1890–1898.

[3] http://www.aliz-e.org

[4] Coninx, A., Baxter, P., Oleari, E., Bellini, S., Bierman, B., Blanson Henkemans, O., Cañamero, L., Cosi, P., Enescu, V., Ros Espinoza, R., Hiolle, A., Humbert, R., Kiefer, B., Kruijff-Korbayová, I., Looije, R., Mosconi, M., Neerincx, M., Paci, G., Patsis, G., Pozzi, C., Sacchitelli, F., Sahli, H. et al. (in press). Towards Long-Term Social Child-Robot Interaction: Using Multi-Activity Switching to Engage Young Users. To appear in *Journal of Human-Robot Interaction*. http://www.coninx.org/work/ALIZE-JHRI-preprint.pdf

[5] Kruijff-Korbayová, I., Oleari, E., Baroni, I., Kiefer, B., Zelati, M. C., Pozzi, C. et al (2014). Effects of Off-Activity Talk in Human-Robot Interaction with Diabetic Children. In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication. (RoMan 2014)*, 649-654.

[6] Nalin, M., Baroni, I., Sanna, A. & Pozzi, C (2012). Robotic Companion for Diabetic Children. In *Proceedings of the 11th International Conference on Interaction Design and Children.* 260-263.

[7] Brooks A. G., Arkin R. C. (2007) Behavioural overlays for non-verbal communication expression on a humanoid robot. *Autonomous robots*. 22/1 55-74.

[8] Nalin, M., Baroni, I., Kruijff-Korbayová, I., Cañamero, L., Lewis, M., Beck, A. et al (2012). Children's adaptation in multi-session interaction. In *Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication* (RoMan 2012).

[9] Baxter, P., Wood, R., Baroni, I., Kennedy, J., Nalin, M. & Belpaeme, T (2013). Emergence of Turn-taking in Unstructured Child-Robot Interactions. In the *Proceedings of the 8th ACM/IEEE International Conference on Human Robot Interaction (HRI)*. 77-78.

An OWL Ontology for Biographical Knowledge. Representing Time-Dependent Factual Knowledge

Hans-Ulrich Krieger and Thierry Declerck

German Research Center for Artificial Intelligence (DFKI), Universität des Saarlandes, Allgemeine Linguistik
krieger@dfki.de, declerck@dfki.de

Abstract

Representing time-dependent information has become increasingly important for reasoning and querying services defined on top of RDF and OWL. In particular, addressing this task properly is vital for practical applications such as modern biographical information systems, but also for the Semantic Web/Web 2.0/Social Web in general. Extending binary relation instances with temporal information often translates into a massive proliferation of useless container objects when trying to keep the underlying RDF model. In this paper, we argue for directly extending RDF triples with further arguments in order to easily represent time-dependent factual knowledge and to allow for practical forms of reasoning. We also report on a freely available lightweight OWL ontology for representing biographical knowledge that models entities of interest via a tri-partite structure of the pairwise disjoint classes Abstract, Object, and Happening. Even though the ontology was manually developed utilizing the *Protégé* ontology editor, and thus sticking to the triple model of RDF, the meta-modelling facilities allowed us to cross-classify all properties as being either synchronic or diachronic. When viewing the temporal arguments as "extra" arguments that only apply to relation instances, universal biographical knowledge from the ontology can still be described as if there is *no* time.

Keywords: OWL biography ontology, representation of time-dependent information, practical temporal reasoning.

1 Synchronic and Diachronic Relations

Linguistics and philosophy make a distinction between synchronic and diachronic relations in order to characterize statements whose truth values do or do not change over time. *Synchronic* relations, such as dateOfBirth, are relations whose instances stay constant over time, thus there is no direct need to attach a temporal extent to them. Consider, e.g., the natural language sentence:

Tony Blair was born on May 6, 1953.

Assuming a RDF-based N-triple representation (Carothers and Seaborne, 2014), an information extraction system might yield the following set of triples:

tb rdf:type Person
tb hasName "Tony Blair"
tb dateOfBirth "1953-05-06"^^xsd:date

Since there is only one unique date of birth, this works perfectly well and properly capture the intended meaning.

Diachronic relationships, however, vary with time, i.e., their truth value do change over time. Representation frameworks such as OWL that are geared towards unary and binary relations can not be extended directly by further (temporal) arguments. Consider the following biographical information:

Christopher Gent was Vodafone's chairman until July 2003. Later, Chris became the chairman of GlaxoSmithKline with effect from January 1st, 2005.

From these two sentences, the information extraction system might discover the following underspecified time-dependent facts:

cg isChairman vf @ [????-??-??, 2003-07-??]
cg isChairman gsk @ [2005-01-01, ????-??-??]

Applying the synchronic representation schema for dateOfBirth from above would give us:

cg isChairman vf
cg holdsAt [????-??-??, 2003-07-??]
cg isChairman gsk
cg holdsAt [2005-01-01, ????-??-??]

However, the association between the original statements and their temporal extents get lost in the resulting RDF graph:

cg isChairman vf @ [????-??-??, 2003-07-??]
~~cg isChairman vf @ [2005-01-01, ????-??-??]~~
~~cg isChairman gsk @ [????-??-??, 2003-07-??]~~
cg isChairman gsk @ [2005-01-01,????-??-??]

as the second and third association are not supported by the above natural language quotation.

2 Approaches for Representing Time-Dependent Knowledge

Several well-known proposals have been presented in the literature in order to equip (binary) relation instances with time or other kinds of information. The individual rewriting schemas are depicted in Figure 1; see Welty and Fikes (2006) and Krieger (2014) for a closer overview:

1. directly equip the relation instance with additional/temporal arguments (Krieger, 2012);
2. apply a meta-logical predicate as used in the situation calculus (McCarthy and Hayes, 1969);
3. reify the original relation à la RDF, turning the property into a class (Manola and Miller, 2004);
4. employ a fact identifier à la YAGO, implicitly leading to quads (Hoffart et al., 2011);
5. wrap the range arguments in an object, called N-ary relation encoding by W3C (Hayes and Welty, 2006);
6. encode a perdurantist/4D view in OWL (Welty and Fikes, 2006);

approach	rewriting schema
1	$\mathit{marriedTo}(p,p') \longmapsto \mathit{marriedTo}(p,p',\underline{s,e})$
2	$\underline{\mathit{holds}}(\mathit{marriedTo}(p,p'),\underline{t}) \longmapsto \exists f\,.\,\mathit{holds}(f,t) \wedge$ $\mathit{type}(f,\mathit{Fluent}) \wedge \mathit{subject}(f,p) \wedge \mathit{predicate}(f,\mathit{marriedTo}) \wedge \mathit{object}(f,p')$
3	$\underline{\mathit{marriedTo}(p,p',s,e)} \longmapsto \exists e\,.\,\mathit{type}(e,\mathit{MarriedToEvent}) \wedge$ $\mathit{person1}(e,p) \wedge \mathit{person2}(e,p') \wedge \mathit{starts}(e,s) \wedge \mathit{ends}(e,e)$
4	$\underline{\mathit{marriedTo}(p,p',s,e)} \longmapsto \exists i\,.\,i := \mathit{marriedTo}(p,p') \wedge \mathit{starts}(i,s) \wedge \mathit{ends}(i,e)$
5	$\underline{\mathit{marriedTo}(p,p',s,e)} \longmapsto \exists o\,.\,\mathit{marriedTo}(p,o) \wedge$ $\mathit{type}(o,\mathit{PersonTime}) \wedge \mathit{person}(o,p') \wedge \mathit{starts}(o,s) \wedge \mathit{ends}(o,e)$
6	$\underline{\mathit{marriedTo}(p,p',s,e)} \longmapsto \exists t,t'\,.\,\mathit{marriedTo}(t,t') \wedge$ $\mathit{type}(t,\mathit{TimeSlice}) \wedge \mathit{hasTimeSlice}(p,t) \wedge \mathit{type}(t',\mathit{TimeSlice}) \wedge \mathit{hasTimeSlice}(p',t') \wedge$ $\mathit{starts}(t,s) \wedge \mathit{ends}(t,e) \wedge \mathit{starts}(t',s) \wedge \mathit{ends}(t',e)$
7	$\underline{\mathit{marriedTo}(p,p',s,e)} \longmapsto \exists t,t'\,.\,\mathit{marriedTo}(t,t') \wedge$ $\mathit{type}(t,\mathit{Person}) \wedge \mathit{hasTimeSlice}(p,t) \wedge \mathit{type}(t',\mathit{Person}) \wedge \mathit{hasTimeSlice}(p',t') \wedge$ $\mathit{starts}(p,s) \wedge \mathit{ends}(p,e) \wedge \mathit{starts}(p',s) \wedge \mathit{ends}(p',e)$
8	$\mathit{marriedTo}(p,p',\underline{s,e}) \longmapsto \mathit{marriedTo\_s\_e}(p,p') \wedge$ $\mathit{marriedTo\_s\_e} \sqsubseteq \mathit{marriedTo} \wedge \mathit{starts}(\mathit{marriedTo\_s\_e},s) \wedge \mathit{ends}(\mathit{marriedTo\_s\_e},e)$
9	$\mathit{marriedTo}(p,p',\underline{s,e}) \longmapsto \mathit{marriedTo}(p,p') \wedge \mathit{starts}(p',s) \wedge \mathit{ends}(p',e)$

Figure 1: Different ways of representing the atemporal statement (the "fluent") $\mathsf{marriedTo}(p,p')$ between two people p and p', being true for the time period $t = [s,e]$. "$\longmapsto$" should be read as *rewrite to*. The last representation schema only works if the original property (here: marriedTo) is inverse functional for all relation instances (which needs not to be the case).

7. interpret the original entities as time slices (Krieger, 2008);
8. encode the temporal extent through new synthetic properties (Gangemi, 2011);
9. use relation composition applied to the second argument which does not work in general, but only if original relation is inverse functional.

2.1 Discussion

The above approaches are in a certain sense semantically equivalent in that we can rewrite one approach to another one without losing any information. It is worth noting that all approaches invalidate standard OWL reasoning, even though they can be implemented within the RDF framework, and thus at least explicitly stated information can be queried by, e.g., SPARQL engines. Nevertheless, the non-temporal entailment rules for RDFS (Hayes, 2004) and OWL Horst/OWL 2 RL (ter Horst, 2005; Motik et al., 2012) can be adjusted, so that rule-based reasoners that go beyond symbol matching, such as *Jena* (Reynolds, 2006) or *HFC* (Krieger, 2013), are still able to perform extended entailments under these new encoding schemas.

Most of the above approaches require to rewrite the original ontology, sometimes by turning relations into classes. With the exception of approach **1**, all approaches require to introduce one or even two brand-new individuals per time-dependent fact (see Figure 1). As a consequence, reasoning and querying with such representations is extremely complex, expensive, and error-prone. Furthermore, the representation schemas **2–7** bear the potential of a non-terminating closure computation in case the newly introduced individuals are viewed as existentially quantified, i.e., anonymous logic variables (RDF: blank nodes). Luckily, this last danger can often be avoided by generating unique URI names that are deterministically generated from their "parts" (i.e., from information that is accessible through properties from the new individual)—this "trick" reminds us of constructing perfect hash functions over complex objects, as known from computer science.

Approach **1** is pursued in the temporal database community under the heading *valid time* (Snodgrass, 2000). The measurements in Krieger (2012) and Krieger (2014) have shown that this approach easily outperforms all other approaches during querying and reasoning (computation of the deductive closure) in the time domain by *several orders of magnitude*. In some cases, this divergency can make a difference between *doable* and *intractable* applications. Consequently, we think the time now is ripe for allowing n-ary relations, or as Schmolze (1989) once put it in the early days of KL-ONE *"... the advantages for allowing direct representation of n-ary relations far outweigh the reasons for the restriction."*

2.2 Tuples vs. Triples: Representation & Reasoning

We would like to make our preference towards a direct representation of additional (temporal) arguments more clear by looking at concrete examples. Consider the Wikipedia entry for Tony Blair which says he married Cherie Booth on 29th March 1980 (today = 2015-05-08), leading to the *quintuple* representation:

tony_blair marriedTo cherie_booth
"1980-03-29"^^xsd:date "2015-05-08"^^xsd:date

A *meaning-preserving triple representation* which adheres to a W3C best practice recommendation, called *N-ary relation encoding* (see rewrite schema **5** in Figure 1) would instead result in five triples, a new individual _:ppt, a new type ValuePlusTime, and the three "accessor" properties hasValue, starts, and ends:

tony_blair marriedTo _:ppt

```
_:ppt rdf:type nary:ValuePlusTime
_:ppt nary:hasValue cherie_booth
_:ppt nary:starts "1980-03-29"^^xsd:date
_:ppt nary:ends "2015-05-08"^^xsd:date
```

Such a representation has a three times larger memory footprint, a slightly more complex structure, and is a bit harder to read. However, as indicated above, the new individual (in our example blank node _:ppt) might turn out to be problematic during entailment reasoning (no longer guaranteed to terminate).

Now let us focus not only on the representation of (static) knowledge, but on the (dynamic) derivation of new knowledge through entailment rules in order to see how much worse a (recommended) triple representation becomes. Consider the following entailment schema for functional diachronic *datatype* properties (in Section 3.3, we will look at the corresponding entailment schema for functional diachronic *object* properties). The original non-temporal schema looks like this (we use the rule syntax of *HFC* (Krieger, 2013) in the examples below):

```
?p rdf:type owl:FunctionalProperty
?p rdf:type owl:DatatypeProperty
?x ?p ?y
?x ?p ?z
→
?x rdf:type owl:Nothing
@test
?y != ?z
```

Such a rule schema is useful, e.g., for detecting contradictory birth dates for one and the same person (famous example: *Louis Armstrong*; right: August 4, 1901, wrongly claimed by him: July 4, 1900). Such a schema matches, for instance,

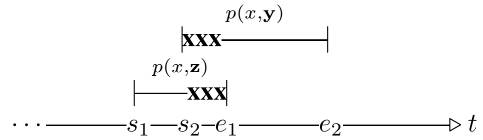
```
louis_armstrong dateOfBirth "1901-08-04"^^xsd:date
louis_armstrong dateOfBirth "1900-07-04"^^xsd:date
```

and binds louis_armstrong to ?x, dateOfBirth to ?p, "1901-08-04"^^xsd:date to ?y, and "1900-07-04"^^xsd:date to ?z. Having found problematic cases is signaled by assigning the "bottom" type owl:Nothing to the subject element of the triple bound to the logical variable ?x (= Louis Armstrong) on the right hand side of the rule.

Adding time to this rule schema makes it applicable to other functional relations such as hasSalary which do change over time, as indicated by the property characteristics time:DiachronicProperty in the rule below. Extending the rule schema is quite easy by equipping the fourth and fifth left hand side clauses with a temporal extent (things that have been added are underlined):

```
?p rdf:type owl:FunctionalProperty
?p rdf:type time:DiachronicProperty
?p rdf:type owl:DatatypeProperty
?x ?p ?y ?s1 ?e1
?x ?p ?z ?s2 ?e2
→
?x rdf:type owl:Nothing ?s ?e
@test
?y != ?z
IntersectionNotEmpty ?s1 ?e1 ?s2 ?e2
@action
?s = Max2 ?s1 ?s2
?e = Min2 ?e1 ?e2
```

The additional left hand side four-place relation IntersectionNotEmpty from the @test section of the rule simply checks whether the two temporal intervals $[s_1, e_1]$ and $[s_2, e_2]$ have a non-empty intersection, indicated by **xxx** below:



If this is the case, we mark the subject bound to ?x being of type owl:Nothing (same as for the original rule), but this type assignment now only holds for the overlapping observation time, given by the maximum of the starting times (= ?s) and the minimum of the ending times (= ?e), as computed in the @action section of the rule.

The above natural extension of the non-temporal rule, however, turns into an awfully looking and terribly inefficient rule when being couched in a triple-based setting:

```
?p rdf:type owl:FunctionalProperty
?p rdf:type time:DiachronicProperty
?p rdf:type owl:DatatypeProperty
?x ?p ?blank1
?blank1 rdf:type nary:ValuePlusTime
?blank1 nary:hasValue ?y
?blank1 nary:starts ?start1
?blank1 nary:ends ?end1
?x ?p ?blank2
?blank2 rdf:type nary:ValuePlusTime
?blank2 nary:hasValue ?z
?blank2 nary:starts ?start2
?blank2 nary:ends ?end2
→
?x rdf:type ?new
?new rdf:type nary:ValuePlusTime
?new nary:hasValue owl:Nothing
?new nary:starts ?start
?new nary:ends ?end
@test
?y != ?z
IntersectionNotEmpty ?start1 ?end1 ?start2 ?end2
@action
?start = Max2 ?start1 ?start2
?end = Min2 ?end1 ?end2
?new = MakeUri owl:Nothing ?start ?end
```

Note how the relevant input information is hidden in the two container individuals bound to ?blank1 and ?blank2 and how the output is wrapped in a brand-new individual ?new, generated by MakeUri from the @action section.

2.3 Limitations

Several points are worth mentioning here. Firstly, we are not dealing here with *duration time* in order to resolve expressions like *Monday* or *20 days* against valid time. This needs to be handled by a richer temporal ontology and temporal arithmetic.

Secondly, *temporal quantification*, such as *four hours every week*, needs to be addressed by a richer temporal inventory. Thirdly, even though *underspecified time* is handled by our implementation through wildcards in the XSD dateTime format (e.g., year missing in *Over New Year's Eve, I have visited the Eiffel Tower*), we do not focus on this here. The solution requires to make certain rule tests sensitive to the fact that underspecified time is only partially ordered. These tests then return *true*, *false*, or *don't-know*, whereas only *true* indicates that the test has succeeded, leading to the instantiation of the right hand side of the rule.

Fourthly, coalescing temporal information (i.e., building larger intervals from intervals with overlapping parts) should be addressed in custom rules and should not be regarded as part of the extended RDFS/OWL rule set, since this functionality depends on the (semantic) nature of predicates and the assumption whether temporal intervals are convex (i.e., contain no "holes") or not.

And finally, certain temporal inferences such as $p(\vec{x}, s, t)$ *entails* $p(\vec{x}, s', t')$ in case $s \leq s' \leq t' \leq t$ should *not* be handled in the below rules, since termination of the computation of the deductive closure is no longer guaranteed. Such information can only be obtained on the query level.

3 Ontology for Biographical Knowledge

We already indicated that we favor approach **1** as it is the most perspicuous of the nine approaches presented above, shows the best memory and runtime footprint, and always guarantees a terminating closure computation for extended RDFS (Hayes, 2004) and OWL (ter Horst, 2005) entailment, as shown in Krieger (2012).

In the introduction, we argued that axiomatic knowledge about classes (TBox) and properties (RBox) does *not* need to have a notion of time—this is universal knowledge which we assume to be *static*. For instance, we do *not* assume that the subtype relationship between two classes only holds for some period of time or that an URI should be regarded as a property at time t and as a class at a different time t' (even though this would be possible). The assertional knowledge of an ontology (ABox), i.e., the set of relation instances, however, is what we equip with time (see the various approaches for the marriedTo example in Figure 1), as this is knowledge that has undergone a temporal change.

In this section, we present the schema (the TBox and the RBox) of an ontology that we had developed originally for the TAKE project (http://take.dfki.de) and that was used in the KOMPARSE project (http://komparse.dfki.de) to represent *biographical information* about celebrities (Adolphs et al., 2010). This ontology has been reused and extended in the EU projects MONNET (http://cordis.europa.eu/fp7/ict/language-technologies) and TRENDMINER (http://www.trendminer-project.eu). This biography ontology is now *part* of a larger set of *independently* developed ontologies (called TMO, for TRENDMINER ONTOLOGIES) which are interlinked to one another through the use of *interface axioms* (Krieger and Declerck, 2014). These interface axioms either relates classes (TBox) and properties (RBox) from different sub-ontologies through the use of description logic axiom constructors, e.g.,

$$\text{bio:Person} \equiv \text{pol:Person}$$

or constrain the domain and range of potentially underspecified properties, e.g.,

$$\top \sqsubseteq \forall \text{op:hasHolder} \;.\; \text{bio:Agent}$$

The property hasHolder from the opinion ontology (prefix op) is a good example of a property for which only the domain has been specified, viz., op:Opinion:

$$\top \sqsubseteq \forall \text{op:hasHolder}^{-} \;.\; \text{op:Opinion}$$

However, hasHolder consciously lacks its range, since this information should only be added when several ontologies are brought together.

The above axioms together with the two terminological axioms from the biography (prefix bio) and the politics (prefix pol) ontologies

$$\text{bio:Person} \sqsubseteq \text{bio:Agent}$$
$$\text{pol:Journalist} \sqsubseteq \text{pol:Person}$$

guarantee to draw legal inferences, such as *journalists are holders of opinions*, even though the interface axiom above constrain holders of opinions to be of type bio:Agent.

TMO has been assembled from 16 sub-ontologies, some of them also dealing with the representation of biographical knowledge, others describing concepts that can be found in politics and sociology. Especially the *opinion ontology* can be used to model *provenance* information, important for biographical knowledge; for instance, information about the:

- holder of the opinion: hasHolder;
- source from which the info was taken: extractedFrom;
- time when the opinion was published: utteredAt;
- trustworthiness of the holder: holdersTrust;
- polarity of the opinion: hasPolarity.

The TMO ontology suite is freely available for academic research and to other sites upon request (see http://www.dfki.de/lt/onto/). Parts of the taxonomic structure of the biography ontology is depicted in Figure 2.

3.1 Overall Guidelines

TMO, and thus the biography ontology, implements several "guidelines" that we have found useful in many projects which have dealt with the representation of time-dependent knowledge (some of the arguments have already been presented):

1. model the TBox and RBox axioms of an ontology as if there is *no* time, since the ontology schema is regarded to be immutable; consequence: standard ontology editors, such as *Protégé* can be used for this task.
2. cross-classify all properties as being either *synchronic* or *diachronic*; advantage: these property characteristics can be used, amongst other things, as distinguishing marks in entailment rules (see examples).
3. populate the ABox of an ontology *with* extended relation instances, i.e., with quintuples whose fourth and fifth argument encode the temporal extent of the preceding atemporal statement (the triple).
4. extend the RDFS/OWL entailment rules by a temporal dimension; example: use XSD's date or dateTime format to implement an interval-based calendar time (used by the examples in this paper).

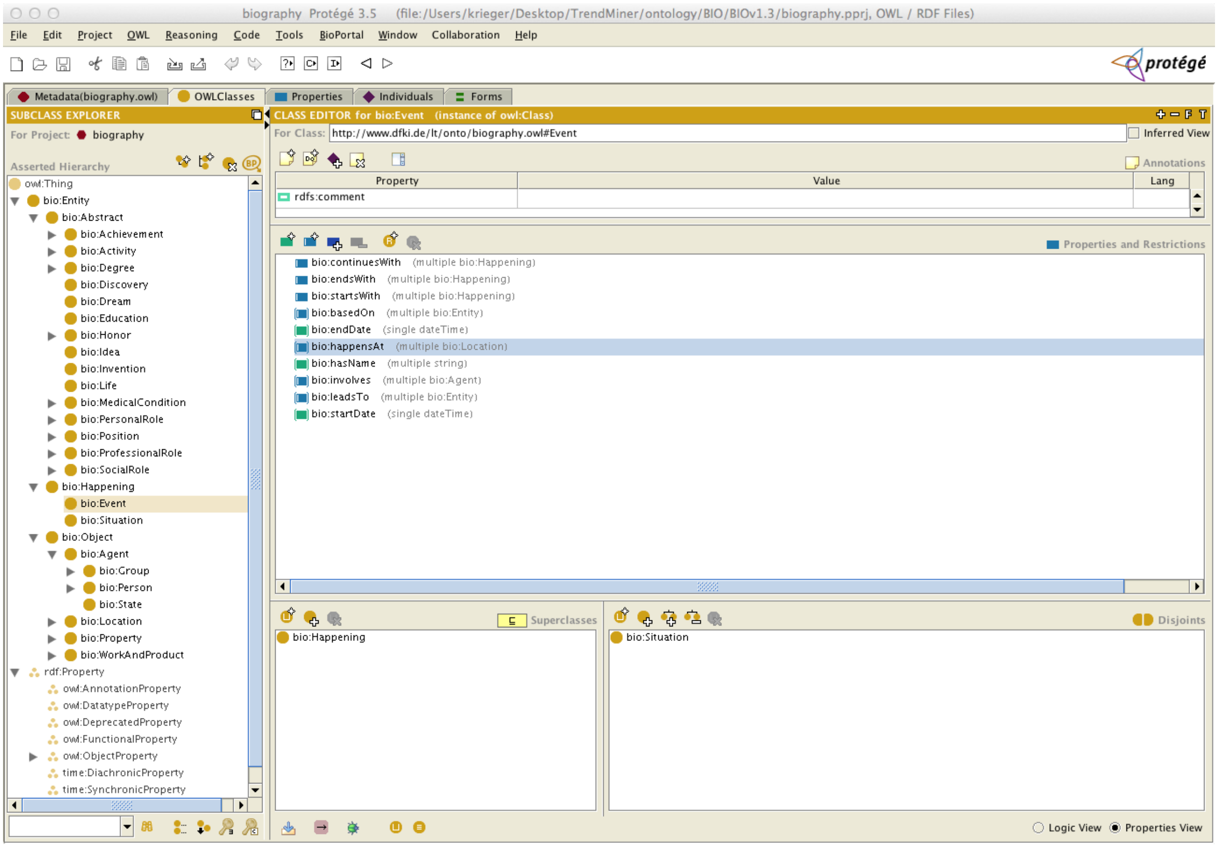


Figure 2: The class subsumption hierarchy of the biography ontology. Note the two subclasses time:DiachronicProperty and time:SynchronicProperty of class rdf:Property that are used to cross-classify (i.e., to type) the properties of the biography ontology; see Figure 4.

3.2 Tri-Partite Structure

The biography ontology assumes a tri-partite structure, defining a most general class Entity, having pairwise disjoint subclasses Abstract, Happening, and Object. TMO is a lightweight ontology that consists of 146 classes and 80 properties, and is of expressivity $\mathcal{SHIN}(D)$, according to the *Ontology metrics* pane of *Protégé*, version 4.3.0. A partial view of the three subclasses and properties linking them is given in Figure 3.

3.2.1 Abstract

Ontological categories that do not fit into Happening or Object are regarded to be of type Abstract, thus this class is a kind of "remainder" category. Abstract things can be used to describe *literal* concepts, e.g., activities, academic degrees, ideas, inventions, the life, or personal, professional, and social roles. An abstraction manifestsIn real-world happenings, whereas the outcome of a happening leadsTo virtually everything (= Entity). For example: a specific military activity (the invasion of Poland) manifested in World War II. The outcome of WW-II has led to military inventions (Abstract), has led to the Cold War (Happening), and has led to the building of 86 U2 aircrafts (Object).

3.2.2 Happening

Happenings are things that "happen" or "unfold" and are disjointly categorized as being either static atomic Situations or dynamic decomposable Events. They come with a (possibly underspecified) startDate and endDate. A happening is basedOn or leadsTo entities (i.e., either abstract things, further happenings, or concrete objects), thus these properties can be used to encode pre- and post-conditions of a happening. An instance of this class also involves Agents and happensAt a Location. Situations help to "terminate" the decomposition of a Happening. The other subclass Event can be used to model simple unordered processes, as it comes with three relational properties of its own, viz., startsWith, continuesWith, and endsWith, all mapping to Happening (see Figure 2).

3.2.3 Object

Objects are "physical" things and mostly deal with Agents (an exhaustive disjoint partition between Person, Group, and political State) and other categories that we think are relevant for biographical information, e.g., Location, material Property, or WorkAndProduct. A Person isAwareOf a Happening: (s)he "owns" it, can be part of it, or learns about a happening. As isAwareOf is a *diachronic* property, awareness of a happening might even turn into oblivion.

3.3 Practical Temporal Reasoning

For a larger non-trivial example, let us again turn our attention to the marriage of *Tony Blair* and *Cherie Booth*. marriedTo is at the same time a *symmetric*, a *diachronic*, a *functional*, and an *object* property (see the *Types* pane at the bottom of Figure 4).

We mentioned that we have cross-classified every property from the biography ontology as being either synchronic

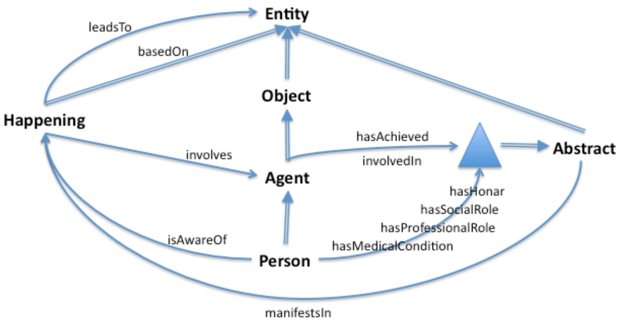


Figure 3: Properties of the biography ontology which relate the three disjoint classes Happening, Object, and Abstract. The solid blue triangle on the right side should indicate subclasses of the class Abstract, such as Achievement.

or diachronic and have already discussed the temporal extension of the entailment rule for functional diachronic *datatype* properties in Section 2.2. Let us now focus on the complementary rule for functional diachronic *object* properties which is applicable to the marriedTo relation:

```
?p rdf:type owl:FunctionalProperty
?p rdf:type time:DiachronicProperty
?p rdf:type owl:ObjectProperty
?x ?p ?y ?s1 ?e1
?x ?p ?z ?s2 ?e2
→
?y owl:sameAs ?z
@test
IntersectionNotEmpty ?s1 ?e1 ?s2 ?e2
```

Here, as in the former example, the additional left hand side test IntersectionNotEmpty checks whether the two temporal intervals $[s_1, e_1]$ and $[s_2, e_2]$ have a non-empty intersection. Assuming that a person is not married to more than one partner at the same time, such a rule is able to identify individuals/URIs bound to ?y and ?z for two properly overlapping observations through the use of owl:sameAs.

Consider again the Wikipedia entry for the marriage of Tony Blair and Cherie Booth that we used in the example from Section 2.2:

```
tony_blair marriedTo cherie_booth
   "1980-03-29"^^xsd:date "2015-05-08"^^xsd:date
```

and furthermore assume that the Economist article *The loneliness of Tony Blair* from December 2014 mentioned that Cherie Blair is Blair's wife (quintuple again):

```
tony_blair marriedTo cherie_blair
   "2014-12-20"^^xsd:date "2014-12-20"^^xsd:date
```

Now it is safe to assume that Cherie Booth and Cherie Blair are in fact the same person, according to the successful application of the above temporal entailment rule:

```
cherie_booth owl:sameAs cherie_blair
```

It is worth noting that sameAs statements will not be equipped with a temporal extent—commonsense dictates that once we do identify individuals, they will never fall apart.

At every moment in time, we never know how long a person is married to his/her partner in advance. That is why we introduced another property divorcedFrom, being the temporal disjoint object property to marriedTo (see the owl:disjointObjectProperty pane in Figure 4). As the Economist article does not specify the date of marriage, we better opt for a moment in time, when Blair and Booth were definitely married (actually a day: start = end). Luckily, the right hand side sameAs inference from above, together with another extended OWL entailment rule, called rdfp11 (ter Horst, 2005), makes sure that even

```
tony_blair marriedTo cherie_blair
   "1980-03-29"^^xsd:date "2015-05-08"^^xsd:date
```

is a valid entailment, exactly what we expect.

3.4 Temporal Arguments as Extra Arguments

So far, our approach has argued for a direct encoding of the temporal extent through two further arguments, turning a binary relation, such as marriedTo ⊆ Person × Person into a quaternary one: marriedTo ⊆ Person × Person × date × date. Given the original relation signature, the *non*-temporal entailment rule schema for *symmetric* binary relations from ter Horst (2005) thus leads to the following instantiation:

$$\textit{marriedTo}(p, p') \rightarrow \textit{marriedTo}(p', p)$$

as *symmetric* relations swap their domain and range arguments (p, p' being two people).

Now, if we add time (b = begin; e = end), we obtain:[1]

$$\textit{marriedTo}(i; j, b, e) \rightarrow \textit{marriedTo}(j, b, e; i)$$

Clearly, something has gone wrong here because symmetric relations assume the same number of arguments in domain and range position. One solution would be to reduplicate the starting and ending points, so we would end up in sexternary relation:

$$\textit{marriedTo}(i, b, e; j, b, e) \rightarrow \textit{marriedTo}(j, b, e; i, b, e)$$

This is *not* an appealing solution as the structures become larger, and rules and queries are harder to formulate, read, debug, and process. What we would like to see is something like:

$$\textit{marriedTo}(i; j; b, e) \rightarrow \textit{marriedTo}(j; i; b, e)$$

whereas the *second* semicolon should indicate that the additional temporal arguments are *extra* arguments, belonging to the relation instance as such (a kind of relation instance annotation, not possible in OWL). Thus with this idea in mind, we can still keep the idea of having only binary relations, *without* introducing any new identifier (contrary to the rewrite schemas **2–7** from Figure 1).

Nevertheless, we are *not* arguing against arbitrary n-ary relations as we are convinced that many binary relations in today's ontologies are ignoring additional arguments (e.g., properties oriented towards ditransitive verbs or having additional modifiers/adjuncts) or come along with unsatisfactory means to encode the additional arguments (relation composition, by taking the object of a binary relation instance into account). The current biography ontology, for instance, poorly models the property obtains as a relation

[1] For better readability, we separate the domain and range arguments from one another by using a semicolon.

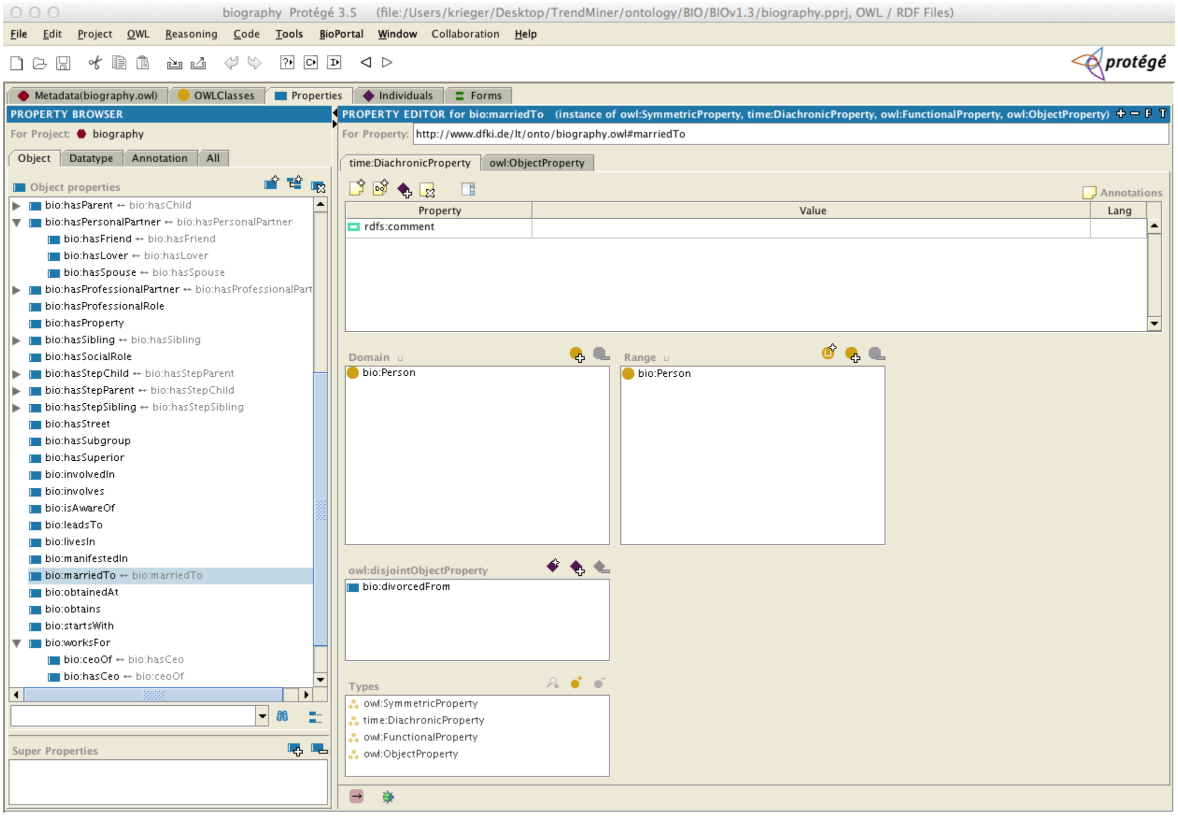


Figure 4: The property subsumption hierarchy of the biography ontology.

between people and (academic) degrees. In order to obtain the educational organization where the degree was obtained, we employ relation composition at the moment, using an additional property obtainedAt between degree and education:

$$\mathsf{obtainedAt} \circ \mathsf{obtains} \subseteq \mathsf{Person} \times \mathsf{EducationalOrganization}$$

This way of representing the additional argument is related to approach **9** from Figure 1 and only works because obtains is *inverse functional* (a characteristics applicable to properties in OWL). Ideally, obtains should be modeled as a quinternary relation, having one domain argument, two range arguments, and two extra temporal arguments:

$$\begin{aligned} \mathsf{obtains} \subseteq \mathsf{Person} \times & & \text{// domain} \\ & \mathsf{Degree} \times \mathsf{EducationalOrganization} \times & \text{// range} \\ & \mathsf{xsd{:}dateTime} \times \mathsf{xsd{:}dateTime} & \text{// extra} \end{aligned}$$

In order to easily define such non-binary relations, ontology editors need to be extended by *Cartesian* types. In Krieger and Willms (2015), we described ×-*Protégé*, an extension of the *Protégé* ontology editor that provides means to define such Cartesian types and to use them to type the domain, range, and extra arguments of non-binary relations. A first public version of ×-*Protégé* will be available in mid 2015.

4 Relation vs. Event Representation

The approaches considered in Section 2 were investigated on how well they perform w.r.t. *binary* relations whose two arguments are considered to be *obligatory*. Such a kind of relation is the default case in today's popular knowledge resources, such as YAGO, DBpedia, BabelNet, or Google's Knowledge Graph.

In case more (e.g., time) and especially *optional* arguments are investigated, our verdict concerning the different approaches might turn into a different direction, so the representation format needs to be updated (in the best case) or changed (in the worst case). Consider the following example, taken from (Davidson, 1967, p. 83):

Jones buttered the toast in the bathroom with a knife at midnight.

The binary base relation butter (we assume a direct mapping of the transitive verb to the relation name here) now needs to be split and/or extended by further optional arguments, as the following sentences are perfectly legal:

Jones buttered the toast.
Jones buttered the toast in the bathroom.
Jones buttered the toast with a knife.
Jones buttered the toast at midnight.
Jones buttered the toast in the bathroom with a knife.
Jones buttered the toast in the bathroom at midnight.
..... etc.

In principle, the number of adjuncts is *not* bounded, thus adding a large number of potentially underspecified direct relation arguments is probably a bad solution. Today's technologies often address such hidden arguments through a kind of *relation composition* as we have seen above for the obtains example from the last section and listed as approach **9** in Figure 1. We think that this modeling "trick" is *unsatisfactory* as it operates on the object of the binary relation instance, but *not* on the relation instance itself (besides being only correct if the original relation is inverse functional, as explained before).

Our *personal* solution would model the *obligatory* arguments, including (under- or unspecified) *time* and perhaps space, as *direct* arguments of the corresponding relation instance or tuple (approach **1**). A further argument, an *event* identifier, also takes part in the relation. Optional arguments, however, would be addressed through binary relations, now working on the *event* argument. Applying this kind of *Davidsonian* or *event* representation to the above example gives us (informal relational notation):

$$\exists e\,.\,\mathit{butter}(e, \mathit{Jones}, \mathit{toast}, \mathit{at\ midnight}) \wedge \\ \mathit{location}(e, \mathit{bathroom}) \wedge \mathit{instrument}(e, \mathit{knife})$$

It is worth noting that two of the approaches from Figure 1 are related to such an event representation, viz., **3** and **4**.

Approach **3** (*internal* reification) can be seen as a kind of "owlfication" of *Neo-Davidsonian* semantics (Parsons, 1990), as the original relation is always turned into an *event* (an OWL class). Here the event identifier e from above directly corresponds to a URI, referring to an instance of the OWL class. For instance, the marriedTo relation is turned into an event class, say Marry; thus:

tony_blair marriedTo cherie_booth
"1980-03-29"^^xsd:date "2015-05-08"^^xsd:date

needs to be expressed by (we use *VerbNet* terminology):

e rdf:type Marry
e agent tony_blair
e co-agent cherie_booth
e starts "1980-03-29"^^xsd:date
e ends "2015-05-08"^^xsd:date

Approach **4** (fact identifier) is a kind of *external* reification. YAGO uses its own extension of the N3 plain triple format, called N4, which associate unique identifiers i with each time-dependent fact. However, the association $i := \mathit{marriedTo}(p, p')$ has the disadvantage of *not* being part of the triple repository, as it is a quadruple technically. So we guess that there exists a separate extendable mapping table outside of the semantic repository, storing the triples.

Luckily, the biography ontology presented in Section 3 both allows for extended relation instances (as shown before), but also Davidsonian-like events through the class Happening and its subclasses Event and Situation (see Figure 2). As there does not exist a Marry event class so far (but only the marriedTo property), such a class needs to be introduced as a subclass of class Event, if needed.

5 Related Ontologies

Several ontologies addressing the representation of biographical information, cultural heritage information, and news-related information exist today, all building on Description Logics and Semantic Web technology standards. These include *ESO* (Segers et al., 2015), *Wikidata* (Erxleben et al., 2014), the *BiographyNet* ontology (Ockeloen et al., 2013), the *BBC Storyline Ontology* (Wilton et al., 2013), *SEM* (van Hage et al., 2011), FRBR_{OO} (Le Bœuf, 2010), *LODE* (Shaw et al., 2009), or *Event-Model-F* (Scherp et al., 2009). Some of these ontologies make use of other resources, such as *WordNet*, *FrameNet*, *Wikipedia*, *SUMO*, *DOLCE*, or *CIDOC CRM*. In order to represent time-dependent knowledge, these approaches always need to stick to an event-like representation in which all information is hidden in an object and time is accessible through properties, similar to approach **3** in Figure 1. None of them are able to encode time as direct arguments of a relation instance (approach **1**). A comparison of some of these event ontologies is presented in (Shaw et al., 2009, section 2) and (van Hage et al., 2011, section 5).

As we have indicated in the beginning of Section 3 (Journalist example), OWL axiom constructors and domain/range restrictions allow us to manually interface our biography ontology with other ontologies, may they be complimentary domain ontologies (*opinion, politics, sociology*), overlapping *biography* event ontologies (see above), or even OWL versions of *upper* ontologies (if desired), such as *DOLCE+DnS* (Gangemi et al., 2002), *SUMO* (Niles and Pease, 2001), or *Cyc* (Reed and Lenat, 2002). For instance, if we would like to interface the *BBC storyline ontology*, the following single axiom suffices:

bio:Happening $\equiv$ nsl:Event

Connecting with *LODE* essentially reduces to:

bio:Happening $\equiv$ lode:Event
bio:happensAt $\equiv$ lode:atPlace
bio:involves $\equiv$ lode:involvedAgent
bio:basedOn $\sqsubseteq$ lode:involved
bio:leadsTo $\sqsubseteq$ lode:involved

Other properties from *LODE* either do not have a direct counterpart (lode:illustrate) or need to be decomposed (lode:atTime onto bio:startDate and bio:endDate). The sub-properties bio:startsWith, bio:continuesWith, and bio:endsWith from the class bio:Event would even allow us to decompose *LODE* events into smaller units, a feature partially available in the *SEM* ontology:

bio:startsWith $\sqsubseteq$ sem:hasSubEvent
bio:continuesWith $\sqsubseteq$ sem:hasSubEvent
bio:endsWith $\sqsubseteq$ sem:hasSubEvent

As our ontology comes with the class bio:Happening, it is possible to take advantage of the great effort invested in the definition of event types in the *ESO* ontology. We finally note that some of the mappings are not expressible through simple OWL axiom constructors, because they involve a translation from n-ary relation instances to sets of triples (and vice versa). This would require to apply *HFC migration rules*, similar to the rewrite rule of approach **3** in Figure 1 which mediates between the quaternary marriedTo relation and its event representation MarriedToEvent.

6 Summary and Conclusion

In this paper, we have presented an overview of nine approaches to the representation of time-dependent knowledge and have favored the direct encoding of the temporal information as extra arguments of the original relation instance. Nevertheless, allowing at the same time for an event-based representation of situations, happening in the real world, is profitable as a knowledge engineer might choose the representation which fits her/his needs. For instance, a marriage ceremony between two people is probably modeled best as an event, whereas the fact that these two people are married for a specific time period is better represented as a quaternary relation. The lightweight biography ontology, presented in this paper, allows both views through the very general class Happening and relations defined between classes which are extended by a starting and ending time, expressing the temporal extent in which the atemporal fact is true (called *valid time* in temporal databases).

Our debate on the right representation format can even be viewed as the more *general quest* on how to integrate/add important (meta) information that has been neglected in the past for practical matters, but has gained a lot of attention recently; see the W3C recommendation for the *provenance* data model *PROV-DM* (Moreau and Missier, 2013). This additional information might include the *holder* of a time-dependent statement or event (person, website, program/service), the spacial *location* of the holder, the *time* when the statement/event was communicated by the holder or made public on the Web (related to *transaction time* in temporal databases), the *trustworthiness* of the holder, and the attitude of the holder w.r.t. the statement/event (sentiment/opinion). Ontologies for all these different aspects already exist today (for instance, the *BiographNet* ontology (Ockeloen et al., 2013) which incorporates a multi-level, multi-perspective model for provenance), but a unified standard is still missing. As a short-/mid-term workaround, we suggest to manually interface these different sources of information, as indicated in Section 5, thus making it possible to incorporate work carried out by other researchers.

Acknowledgements

The work described in this paper has been partially carried out in the *ALL SIDES* project, funded by the German Federal Ministry of Education and Research under contract number FKZ 01IW14002 and the *PHEME* project, funded by the European Union's 7th Framework Programme under grant agreement no 611233. The authors would like to thank our three reviewers and Antske Fokkens for their useful comments, Hans Uszkoreit for giving the initial impetus for the development of the biography ontology several years ago, and Bernd Kiefer for always having an open ear in many discussions.

7 References

Peter Adolphs, Xiwen Cheng, Tina Klüwer, Hans Uszkoreit, and Feiyu Xu. 2010. Question answering biographic information and social network powered by the semantic web. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 2764–2768.

Gavin Carothers and Andy Seaborne. 2014. RDF 1.1 N-Triples. a line-based syntax for an RDF graph. Technical report, W3C.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing Wikidata to the Linked Data Web. In *Proceedings of the 13th International Semantic Web Conference (ISWC)*, pages 50–65.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. 2002. Sweetening ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 166–181.

Aldo Gangemi. 2011. SuperDuper schema: an OWL2+RIF DnS pattern. In *KCAP 2011* Deep Knowledge Representation Challenge Workshop.

Patrick Hayes and Chris Welty. 2006. Defining N-ary relations on the Semantic Web. Technical report, W3C.

Patrick Hayes. 2004. RDF semantics. Technical report, W3C.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International World Wide Web Conference (WWW 2011)*, pages 229–232.

Hans-Ulrich Krieger and Thierry Declerck. 2014. TMO—the federated ontology of the TrendMiner project. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC)*.

Hans-Ulrich Krieger and Christian Willms. 2015. Extending OWL ontologies by Cartesian types to represent N-ary relations in natural language. In *Proceedings of the IWCS Workshop on Language and Ontologies*.

Hans-Ulrich Krieger. 2008. Where temporal description logics fail: Representing temporally-changing relationships. In *KI 2008: Advances in Artificial Intelligence*, volume 5243 of *Lecture Notes in Artificial Intelligence*, pages 249–257. Springer.

Hans-Ulrich Krieger. 2012. A temporal extension of the Hayes/ter Horst entailment rules and an alternative to W3C's n-ary relations. In *Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS)*, pages 323–336.

Hans-Ulrich Krieger. 2013. An efficient implementation of equivalence relations in OWL via rule and query rewriting. In *Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC)*, pages 260–263.

Hans-Ulrich Krieger. 2014. A detailed comparison of seven approaches for the annotation of time-dependent factual knowledge in RDF and OWL. In *Proceedings of the 10th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA)*.

Patrick Le Bœuf. 2010. From FRBR to FRBR_{OO} through CIDOC CRM. Slide presentation, October. International Symposium on the Future of Information Organization Research, Taipei, Taiwan.

Frank Manola and Eric Miller. 2004. RDF primer. Technical report, W3C.

John McCarthy and Patrick J. Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press.

Luc Moreau and Paolo Missier. 2013. PROV-DM: The

PROV data model. Technical report, W3C. W3C Recommendation 30 April 2013.

Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. 2012. OWL 2 web ontology language profiles. Technical report, W3C. W3C Recommendation 11 December 2012.

Ian Niles and Adam Pease. 2001. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS)*, pages 2–9.

Niels Ockeloen, Antske Fokkens, Serge ter Braake, Piek Vossen, Victor de Boer, Guus Schreiber, and Susan Legêne. 2013. BiographyNet: Managing provenance at multiple levels and from different perspectives. In *Proceedings of the 3rd International Workshop on Linked Science (LISC) at ISWC*, pages 59–71.

Terence Parsons. 1990. *Events in the Semantics of English. A Study in Subatomic Semantics*. MIT Press, Cambridge, MA.

Stephen L. Reed and DouglaS B. Lenat. 2002. Mapping ontologies into Cyc. In *Proceedings of the AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, pages 1–6.

Dave Reynolds. 2006. Jena rules tutorial. In *Jena User Conference*. PowerPoint presentation.

Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. 2009. F—a model of events based on the foundational ontology DOLCE+DnS Ultralite. In *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP)*, pages 137–144.

James G. Schmolze. 1989. Terminological knowledge representation systems supporting n-ary terms. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 432–443.

Roxane Segers, Piek Vossen, Marco Rospocher, Luciano Serafini, Egoitz Laparra, and German Rigau. 2015. ESO: a frame based ontology for events and implied situations. In *Proceedings of MAPLEX*.

Ryan Shaw, Raphaël Troncy, and Lynda Hardman. 2009. LODE: Linking open descriptions of events. In *Proceedings of the 4th Asian Semantic Web Conference (ASWC)*, pages 153–167.

Richard T. Snodgrass. 2000. *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann, San Francisco, CA.

Herman J. ter Horst. 2005. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *Journal of Web Semantics*, 3:79–115.

Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. 2011. Design and use of the simple event model (SEM). *Journal of Web Semantics*, 9(2):128–136.

Christopher Welty and Richard Fikes. 2006. A reusable ontology for fluents in OWL. In *Proceedings of 4th FOIS*, pages 226–236.

Paul Wilton, Jeremy Tarling, and Jarred McGinnis. 2013. Storyline ontology: An ontology to represent news storylines. http://www.bbc.co.uk/ontologies/storyline, May. Contributors: Paul Rissen, Helen Lippell, Matt Chadburn, Tom Leitch, Dan Brickley, Michael Smethurst, Sebastien Cevey.

A Modal Representation of Graded Medical Statements

Hans-Ulrich Krieger[1] and Stefan Schulz[2]

[1] German Research Center for Artificial Intelligence (DFKI) krieger@dfki.de
[2] Institute of Medical Informatics, Medical University of Graz
stefan.schulz@medunigraz.at

Abstract. Medical natural language statements uttered by physicians are usually *graded*, i.e., are associated with a degree of uncertainty about the validity of a medical assessment. This uncertainty is often expressed through specific *verbs*, *adverbs*, or *adjectives* in natural language. In this paper, we look into a *representation* of such graded statements by presenting a simple non-standard *modal logic* which comes with a set of modal operators, directly associated with the words indicating the uncertainty and interpreted through *confidence intervals* in the model theory. We complement the model by a set of RDFS-/OWL 2 RL-like entailment (*if-then*) rules, acting on the syntactic representation of modalized statements. Our interest in such a formalization is related to the use of OWL as the *de facto* standard in (medical) ontologies today and its weakness to represent and reason about assertional knowledge that is uncertain or that changes over time. The approach is not restricted to medical statements, but is applicable to other graded statements as well.

1 Introduction & Background

Medical natural language statements uttered by physicians or other health professionals and found in medical examination letters are usually *graded*, i.e., are associated with a degree of uncertainty about the validity of a medical assessment. This uncertainty is often expressed through specific verbs, adverbs, or adjectives in natural language (which we will call *gradation words*). E.g., *Dr. X suspects that Y suffers from Hepatitis* or *The patient probably has Hepatitis* or *(The diagnosis of) Hepatitis is confirmed*.
In this paper, we look into a representation of such graded statements by presenting a simple non-standard modal logic which comes with a small set of partially-ordered modal operators, directly associated with the words indicating the uncertainty and interpreted through confidence intervals in the model theory. The approach currently only addresses modalized propositional formulae in negation normal form which can be seen as a canonical representation of natural language sentences of the above form (a kind of a *controlled natural language*). Our interest in such a formalization is related to the use of OWL in our projects as the *de facto* standard for (medical) ontologies today and its weakness to represent and reason about assertional knowledge that is uncertain [17] or that

changes over time [14]. There are two principled ways to address such a restriction: *either* by sticking with the existing formalism (viz., OWL) and trying to find an encoding that still enables some useful forms of reasoning [17]; *or* by deviating from a defined standard in order to arrive at an easier, intuitive, and less error-prone representation [14].
Here, we follow the latter avenue, but employ and extend the standard entailment rules from [8] and [21] for positive binary relation instances in RDFS and OWL towards modalized n-ary relation instances, including negation. These entailment rules talk about, e.g., subsumption, class membership, or transitivity, and have been found useful in many applications. The proposed solution has been implemented in *HFC* [15], a forward chaining engine that builds Herbrand models which are compatible with the open-world view underlying OWL. The approach presented in this paper is clearly not restricted to medical statements, but is applicable to other graded statements as well (including *trust*), e.g., technical diagnosis (*the engine is probably overheated*) or more general in everyday conversation (*I'm pretty sure that X has signed a contract with Y*) which can be seen as the common case (contrary to true *universal* statements).

2 Graded Medical Statements: OWL vs. Modalized Representation

We note here that our initial modal operators were inspired by the *qualitative information parts* of diagnostic statements from [17] shown in Figure 1, but we might have chosen other operators, capturing the meaning of the gradation words used in the examples at the beginning of Section 1 (e.g., *probably*).

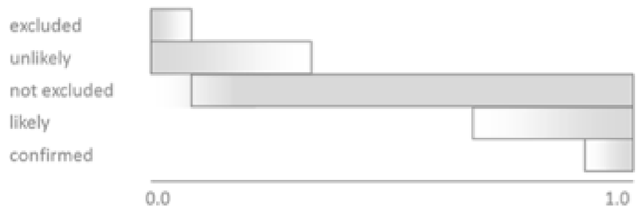


Fig. 1. Vague schematic mappings of the qualitative information parts *excluded* (E), *unlikely* (U), *not excluded* (N), *likely* (L), and *confirmed* (C) to *confidence intervals*, as used in this paper. Figure taken from [17].

These qualitative parts were used in statements about, e.g., liver inflammation with varying levels of detail. From this, we want to infer that, e.g., **if** *Hepatitis is confirmed* **then** *Hepatitis is likely* but **not** *Hepatitis is unlikely*. And **if** *Viral Hepatitis B is confirmed*, **then** both *Viral Hepatitis is confirmed* **and** *Hepatitis is confirmed* (generalization). Things "turn around" when we look at the adjectival modifiers *excluded* and *unlikely*: **if** *Hepatitis is excluded* **then** *Hepatitis is unlikely*, but **not** *Hepatitis is not excluded*. Furthermore, **if** *Hepatitis is excluded*,

then both *Viral Hepatitis is excluded* **and** *Viral Hepatitis B is excluded* (specialization). The set of *plausible* entailments for this kind of graded reasoning is depicted in Figure 2.

		Being said to have hepatitis (H) / viral hepatitis (vH) / viral hepatitis B (vHB) is…														
Precondition:		confirmed			likely			not excluded			unlikely			excluded		
Entailment:		H	vH	vHB	H	vH	vHB	H	vH	vHB	H	vH	vHB	H	vH	vHB
confirmed	H	x	x	x												
	vH		x	x												
	vHB			x												
likely	H	x	x	x	x	x	x									
	vH		x	x		x	x									
	vHB			x			x									
not excluded	H	x	x	x	x	x	x	x	x	x						
	vH		x	x		x	x		x	x						
	vHB			x			x			x						
unlikely	H										x			x		
	vH										x	x		x	x	
	vHB										x	x	x	x	x	x
excluded	H													x		
	vH													x	x	
	vHB													x	x	x

Fig. 2. Statements about liver inflammation with varying levels of detail: *Viral Hepatitis B* (vHB) implies *Viral Hepatitis* (vH) which implies *Hepatitis* (H). The matrix depicts entailments considered plausible, based on the inferences that follow from Figure 1. Hepatitis and its subclasses can be easily replaced by other medical situations/diseases. Figure taken from [17].

[17] consider five encodings (one outside the expressivity of OWL), from which only two were able to fully reproduce the inferences from Figure 2. Let us quickly look on approach 1, called *existential restriction*, before we informally present its modal counterpart (we will use abstract description logic syntax here [2]):

$\mathsf{HepatitisSituation} \equiv \mathsf{ClinicalSituation} \sqcap \exists \mathsf{hasCondition.Hepatitis}$

% *Hepatitis subclass hierarchy*
$\mathsf{ViralHepatitisB} \sqsubseteq \mathsf{ViralHepatitis} \sqsubseteq \mathsf{Hepatitis}$

% *vagueness via two subclass hierarchies*
$\mathsf{IsConfirmed} \sqsubseteq \mathsf{IsLikely} \sqsubseteq \mathsf{IsNotExcluded} \qquad \mathsf{IsExcluded} \sqsubseteq \mathsf{IsUnlikely}$

% *a diagnostic statement about Hepatitis*
$\mathsf{BeingSaidToHaveHepatitisIsConfirmed} \equiv \mathsf{DiagnosticStatement} \sqcap$
$\quad \forall \mathsf{hasCertainty.IsConfirmed} \sqcap \exists \mathsf{isAboutSituation.HepatitisSituation}$

Standard OWL reasoning under this representation then ensures that, for instance,

$\mathsf{BeingSaidToHaveHepatitsIsConfirmed} \sqsubseteq \mathsf{BeingSaidToHaveHepatitisIsLikely}$

is the case, exactly one of the plausible inferences from Figure 2.
The encodings in [17] were quite cumbersome as the primary interest was to stay within the limits of the underlying calculus (OWL). Besides coming up with complex encodings, only minor forms of reasoning were possible, viz., subsumption reasoning. These disadvantages are a result of two conscious decisions:

OWL only provides unary and binary relations (concepts and roles) and comes up with a (mostly) fixed set of entailment/tableaux rules.
In our approach, however, the *qualitative information parts* from Figure 1 are first class citizens of the object language (the modal operators) and *diagnostic statements* from the Hepatitis use case are expressed through the binary property $\mathsf{suffersForm}$ between p (patients, people) and d (diseases, diagnoses). The plausible inferences are then simply a *byproduct* of the *instantiation* of the entailment rule schemas (G) from Section 5.1, and (S1) and (S0) from Section 5.2 for property $\mathsf{suffersForm}$ (the rule variables are universally quantified; $\top$ = *universal truth*; C = *confirmed*; L = *likely*), e.g.,

(S1) $\top\mathsf{ViralHepatitisB}(d) \wedge \mathsf{ViralHepatitisB} \sqsubseteq \mathsf{ViralHepatitis} \rightarrow \top\mathsf{ViralHepatitis}(d)$
(G) $C\mathsf{suffersFrom}(p, d) \rightarrow L\mathsf{suffersFrom}(p, d)$

Two things are worth to be mentioned here. *Firstly*, not only OWL-like properties (binary relations) can be graded, such as $C\mathsf{suffersFrom}(p, d)$ (= *it is confirmed that p suffers from d*), but also class membership (unary relations), e.g., $C\mathsf{ViralHepatitisB}(d)$ (= *it is confirmed that d is Viral Hepatitis B*). However, as the original OWL example above is unable to make use of any modals, we employ a special modal $\top$ here:$\top\mathsf{ViralHepatitisB}(d)$. *Secondly*, modal operators are only applied to assertional knowledge, involving individuals (the ABox in OWL)—neither axioms about classes (TBox) nor properties (RBox) are being affected by modals, as they are supposed to express universal truth.

3 Confidence of Statements and Confidence Intervals

We address the *confidence* of an asserted medical statement [17] through *graded* modalities applied to propositional formulae: E (*excluded*), U (*unlikely*), N (*not excluded*), L (*likely*), and C (*confirmed*). For various (technical) reasons, we add a *wildcard* modality ? (*unknown*), a complementary *failure* modality ! (*error*), plus two further modalities to syntactically state definite truth and falsity: $\top$ (*true*) and $\bot$ (*false*). Let $\triangle$ now denotes the set of all modalities:

$$\triangle = \{?, !, \top, \bot, E, U, N, L, C\}$$

A *measure function*

$$\mu : \triangle \mapsto [0, 1] \times [0, 1]$$

is a mapping which returns the associated *confidence interval* $[l, h]$ for a modality from $\triangle$ ($l \leq h$). We presuppose that

- $\mu(?) = [0, 1]$
- $\mu(!) = \emptyset$[3]
- $\mu(\top) = [1, 1]$
- $\mu(\bot) = [0, 0]$

In addition, we define two disjoint subsets of $\triangle$, called

- $\mathsf{1} = \{\top, C, L, N\}$
- $\mathsf{0} = \{\bot, E, U\}$

[3] Recall that an interval is a set of real numbers, together with a total ordering relation (e.g., $\leq$) over the elements, thus $\emptyset$ is a perfect, although degraded interval.

and again make a presupposition: the confidence intervals for modals from 1 *end* in 1, whereas the confidence intervals for 0 modals always *start* with 0. It is worth noting that we do *not* make use of μ in the syntax of the modal language (for which we employ the modalities from $\triangle$), but in the semantics when dealing with the satisfaction relation of the model theory (see Section 4).

We have talked about *confidence intervals* now several times without saying what we actually mean by this. Suppose that a physician says that it is *confirmed* (= C) that patient p suffers from disease d, given a set of recognized symptoms $S = \{s_1, \ldots, s_k\}$: $C\,\mathit{suffersFrom}(p, d)$.

Assuming that a different patient p' shows the same symptoms S (and only S, and perhaps further symptoms which are, however, *independent* from S), we would assume that the same doctor would diagnose $C\,\mathit{suffersFrom}(p', d)$.

Even an other, but similar trained physician is supposed to grade the two patients *similarly*. This similarity which originates from patients showing the same symptoms and from physicians being taught at the same medical school is addressed by confidence *intervals* and not through a *single* (posterior) probability, as there are still variations in diagnostic capacity and daily mental state of the physician. By using intervals (instead of single values), we can usually reach a consensus among people upon the *meaning* of gradation words, even though the low/high values of the confidence interval for, e.g., *confirmed* might depend on the context.

Being a bit more theoretic, we define a *confidence interval* as follows. Assume a *Bernoulli experiment* [13] that involves a large set of n patients P sharing the same symptoms S. W.r.t. our example, we would like to know whether $\mathit{suffersFrom}(p, d)$ or $\neg\mathit{suffersFrom}(p, d)$ is the case for every patient $p \in P$, sharing S. Given a Bernoulli trials sequence $\boldsymbol{X} = \langle x_1, \ldots, x_n \rangle$ with indicator random variables $x_i \in \{0, 1\}$ for a patient sequence $\langle p_1, \ldots, p_n \rangle$, we can *approximate* the *expected value* E for *suffersFrom* being *true*, given disease d and background symptoms S by the *arithmetic mean* A:

$$\mathrm{E}[\boldsymbol{X}] \approx \mathrm{A}[\boldsymbol{X}] = \frac{\sum_{i=1}^{n} x_i}{n}$$

Due to the *law of large numbers*, we expect that if the number of elements in a trials sequence goes to infinity, the arithmetic mean will coincide with the expected value:

$$\mathrm{E}[\boldsymbol{X}] = \lim_{n \to \infty} \frac{\sum_{i=1}^{n} x_i}{n}$$

Clearly, the arithmetic mean for each new *finite* trials sequence is different, but we can try to *locate* the expected value within an interval around the arithmetic mean:

$$\mathrm{E}[\boldsymbol{X}] \in [\mathrm{A}[\boldsymbol{X}] - \epsilon_1, \mathrm{A}[\boldsymbol{X}] + \epsilon_2]$$

For the moment, we assume $\epsilon_1 = \epsilon_2$, so that $\mathrm{A}[\boldsymbol{X}]$ is in the center of this interval which we will call from now on *confidence interval*.

Coming back to our example and assuming $\mu(C) = [0.9, 1]$, $C\,\mathit{suffersFrom}(p, d)$ can be read as being true in 95% of all cases *known* to the physician, involving patients p potentially having disease d and sharing the same prior symptoms (evidence) $s_1, \ldots, s_k$:

$$\frac{\sum_{p \in P} \mathrm{Prob}(\mathit{suffersFrom}(p,d)|s_1,\ldots,s_k)}{n} \approx 0.95$$

The variance of $\pm 5\%$ is related to varying diagnostic capabilities between (comparative) physicians, daily mental form, undiscovered important symptoms or examinations which have not been carried out (e.g., lab values), or perhaps even the physical stature of the patient which unconsciously affects the final diagnosis, etc, as elaborated above. Thus the individual modals from $\triangle$ express (via μ) different forms of the physician's confidence, depending on the set of already acquired symptoms as (potential) explanations for a specific disease.

4 Model Theory and Negation Normal Form

Let $\mathcal{C}$ denote the set of constants that serve as the arguments of a relation instance. In order to define basic n-ary propositional formulae (ground atoms, propositional letters), let $p(\boldsymbol{c})$ abbreviates $p(c_1, \ldots, c_n)$, for some $c_1, \ldots, c_n \in \mathcal{C}$, given $\mathit{length}(\boldsymbol{c}) = n$. In case the number of arguments do not matter, we sometimes simply write p, instead of, e.g., $p(c,d)$ or $p(\boldsymbol{c})$. As before, we assume $\triangle = \{?, !, \top, \bot, E, U, N, L, C\}$. We inductively define the set of *well-formed formulae* ϕ of our modal language as follows:

$$\phi ::= p(\boldsymbol{c}) \mid \neg\phi \mid \phi \wedge \phi' \mid \phi \vee \phi' \mid \triangle\phi$$

4.1 Simplification and Normal Form

We now syntactically *simplify* the set of well-formed formulae ϕ by restricting the uses of *negation* and *modalities* to the level of propositional letters p and call the resulting language $\mathit{\Lambda}$:

$$\pi ::= p(\boldsymbol{c}) \mid \neg p(\boldsymbol{c})$$

$$\phi ::= \pi \mid \triangle\pi \mid \phi \wedge \phi' \mid \phi \vee \phi' \mid$$

To do so, we need the notion of a *complement* modal δ^{C} for every $\delta \in \triangle$, where

$$\mu(\delta^{\mathsf{C}}) := \mu(\delta)^{\mathsf{C}} = \mu(?) \setminus \mu(\delta) = [0,1] \setminus \mu(\delta)$$

I.e., $\mu(\delta^{\mathsf{C}})$ is defined as the complementary interval of $\mu(\delta)$ (within the bounds of $[0,1]$, of course). For example, E and N (*excluded, not excluded*) or ? and ! (*unknown, error*) are already existing complementary modals. We also require *mirror* modals δ^{M} for every $\delta \in \triangle$ whose confidence interval $\mu(\delta^{\mathsf{M}})$ is derived by "mirroring" $\mu(\delta)$ to the opposite site of the confidence interval, either to the left or to the right:

$$\textbf{if } \mu(\delta) = [l,h] \textbf{ then } \mu(\delta^{\mathsf{M}}) := [1-h, 1-l]$$

For example, E and C (*excluded, confirmed*) or $\top$ and $\bot$ (*top, bottom*) are mirror modals. In order to transform ϕ into its *negation normal form*, we need to apply simplification rules a finite number of times (until rules are no longer applicable). We depict those rules by using the $\vdash$ relation, read as *formula* $\vdash$ *simplified formula*:

1. $?\phi \vdash \epsilon$ % *$?\phi$ is not informative at all, but its existence should alarm us*
2. $\neg\neg\phi \vdash \phi$
3. $\neg(\phi \wedge \phi') \vdash \neg\phi \vee \neg\phi'$
4. $\neg(\phi \vee \phi') \vdash \neg\phi \wedge \neg\phi'$
5. $\neg\triangle\phi \vdash \triangle^{\mathsf{C}}\phi$ (example: $\neg E\phi = N\phi$)
6. $\triangle\neg\phi \vdash \triangle^{\mathsf{M}}\phi$ (example: $E\neg\phi = C\phi$)

Clearly, the mirror modals δ^{M} are not necessary as long as we explicitly allow for negated statements, and thus case 6 can, in principle, be dropped.
What is the result of simplifying $\triangle(\phi \wedge \phi')$ and $\triangle(\phi \vee \phi')$? Let us start with the former case and consider as an example the statement about an engine that *a mechanical failure m and an electrical failure e is confirmed*: $C(m \wedge e)$. It seems plausible to simplify this expression to $Cm \wedge Ce$. Commonsense tells us furthermore that neither Em nor Ee is compatible with this description.
Now consider the "opposite" statement $E(m \wedge e)$ which must *not* be rewritten to $Em \wedge Ee$, as *either Cm or Ce* is well *compatible* with $E(m \wedge e)$. Instead, we rewrite this kind of "negated" statement as $Em \vee Ee$, and this works fine with either Cm or Ce.
In order to address the other modal operators, we generalize these plausible inferences by making a distinction between $\mathsf{0}$ and $\mathsf{1}$ modals (see Section 3):

7a. $\mathsf{0}(\phi \wedge \phi') \vdash \mathsf{0}\phi \vee \mathsf{0}\phi'$
7b. $\mathsf{1}(\phi \wedge \phi') \vdash \mathsf{1}\phi \wedge \mathsf{1}\phi'$

Now let us consider disjunction inside the scope of a modal operator. As we do allow for the full set of Boolean operators, we are allowed to deduce

8. $\triangle(\phi \vee \phi') \vdash \triangle(\neg(\neg(\phi \vee \phi'))) \vdash \triangle(\neg(\neg\phi \wedge \neg\phi')) \vdash \triangle^{\mathsf{M}}(\neg\phi \wedge \neg\phi')$

This is, again, a conjunction, so we apply schemas 7a and 7b, giving us

8a. $\mathsf{0}(\phi\vee\phi') \vdash \mathsf{0}^{\mathsf{M}}(\neg\phi\wedge\neg\phi') \vdash \mathsf{1}(\neg\phi\wedge\neg\phi') \vdash \mathsf{1}\neg\phi\wedge\mathsf{1}\neg\phi' \vdash \mathsf{1}^{\mathsf{M}}\phi\wedge\mathsf{1}^{\mathsf{M}}\phi' \vdash \mathsf{0}\phi\wedge\mathsf{0}\phi'$
8b. $\mathsf{1}(\phi\vee\phi') \vdash \mathsf{1}^{\mathsf{M}}(\neg\phi\wedge\neg\phi') \vdash \mathsf{0}(\neg\phi\wedge\neg\phi') \vdash \mathsf{0}\neg\phi\vee\mathsf{0}\neg\phi' \vdash \mathsf{0}^{\mathsf{M}}\phi\vee\mathsf{0}^{\mathsf{M}}\phi' \vdash \mathsf{1}\phi\vee\mathsf{1}\phi'$

Note how the modals from $\mathsf{0}$ in 7a and 8a act as a kind of *negation* to turn the logical operators into their counterparts, similar to de Morgan's law.

4.2 Model Theory

In the following, we extend the standard definition of modal (Kripke) frames and models [3] for the *graded* modal operators from $\triangle$ by employing the measure function μ and focussing on the minimal definition for ϕ in Λ. A *frame* $\mathcal{F}$ for the probabilistic modal language Λ is a pair

$$\mathcal{F} = \langle \mathcal{W}, \triangle \rangle$$

where $\mathcal{W}$ is a non-empty set of *worlds* (or *situations, states, points, vertices*) and $\triangle$ a family of binary relations over $\mathcal{W} \times \mathcal{W}$, called *accessibility relations*. Note that we have overloaded $\triangle$ (and each $\delta \in \triangle$) in that it refers to the modals used in the syntax of Λ, but also to depict the binary relations, connecting worlds.

A *model* $\mathcal{M}$ for the probabilistic modal language Λ is a triple

$$\mathcal{M} = \langle \mathcal{F}, \mathcal{V}, \mu \rangle$$

such that $\mathcal{F}$ is a *frame*, $\mathcal{V}$ a *valuation*, assigning each proposition ϕ a subset of $\mathcal{W}$, viz., the set of worlds in which ϕ holds, and μ a mapping, returning the confidence interval for a given modality from $\triangle$. Note that we only require a definition for μ in $\mathcal{M}$ (the model, but *not* in the frame), as $\mathcal{F}$ represent the relational structure without interpreting the edge labeling (the modal names) of the graph.
The *satisfaction relation* $\models$, given a model $\mathcal{M}$ and a specific world w is inductively defined over the set of well-formed formulae of Λ in *negation normal form* (remember $\pi ::= p(\boldsymbol{c}) \mid \neg p(\boldsymbol{c})$):

1. $\mathcal{M}, w \models p(\boldsymbol{c})$ **iff** $w \in \mathcal{V}(p(\boldsymbol{c}))$ **and** $w \notin \mathcal{V}(\neg p(\boldsymbol{c}))$
2. $\mathcal{M}, w \models \neg p(\boldsymbol{c})$ **iff** $w \in \mathcal{V}(\neg p(\boldsymbol{c}))$ **and** $w \notin \mathcal{V}(p(\boldsymbol{c}))$
3. $\mathcal{M}, w \models \phi \wedge \phi'$ **iff** $\mathcal{M}, w \models \phi$ **and** $\mathcal{M}, w \models \phi'$
4. $\mathcal{M}, w \models \phi \vee \phi'$ **iff** $\mathcal{M}, w \models \phi$ **or** $\mathcal{M}, w \models \phi'$
5. **for all** $\delta \in \triangle$: $\mathcal{M}, w \models \delta\pi$ **iff** $\frac{\#\{u \mid (w,u) \in \delta \textbf{ and } \mathcal{M}, u \models \pi\}}{\#\{u \mid (w,u) \in \delta' \textbf{ and } \delta' \in \triangle\}} \in \mu(\delta)$

The last case of the satisfaction relation addresses the modals: for a world w, we look for the successor states u that are directly reachable via δ and in which π holds, and divide the number of such states by the number of all worlds that are directly reachable from w. This number between 0 and 1 must lie in the confidence interval $\mu(\delta)$ of δ in order to satisfy $\delta\pi$, given $\mathcal{M}, w$.
It is worth noting that the satisfaction relation above differs in its handling of $\mathcal{M}, w \models \neg p(\boldsymbol{c})$, as negation is *not* interpreted through the *absence* of $p(\boldsymbol{c})$ ($\mathcal{M}, w \not\models p(\boldsymbol{c})$), but through the *existence* of $\neg p(\boldsymbol{c})$. This treatment addresses the *open-world* nature in OWL and the evolvement of a (medical) domain over time.
We also note that the definition of the satisfaction relation for modalities (last clause) is related to the *possibility operators* $M_k \cdot$ ($= \Diamond^{\geq k} \cdot$; $k \in \mathbb{N}$) [6] and *counting modalities* $\cdot \geq n$ [1], used in modal logic characterizations of *description logics* with *cardinality* restrictions.

4.3 Well-Behaved Frames

As we will see later, it is handy to assume that the graded modals are arranged in a kind of hierarchy—the more we move "upwards" in the hierarchy, the more a statement in the scope of a modal becomes *uncertain*. In order to address this, we slightly extend the notion of a *frame* by a third component $\preceq \subseteq \triangle \times \triangle$, a partial order between modalities:

$$\mathcal{F} = \langle \mathcal{W}, \triangle, \preceq \rangle$$

Let us consider the following modal hierarchy that we build from the set $\triangle$ of already introduced modals:

$$\begin{tikzcd}[column sep=small]
 & \top \arrow[r] & C \arrow[r] & L \arrow[r] & N \arrow[dr] & \\
! \arrow[ur] \arrow[dr] & & & & & ? \\
 & \bot \arrow[r] & E \arrow[r] & U \arrow[urr] & &
\end{tikzcd}$$

This graphical representation is just a compact way to specify a set of 33 binary relation instances over $\triangle$, such as, e.g., $\top \preceq \top$, $\top \preceq N$, $C \preceq N$, $\bot \preceq ?$, or $! \preceq ?$. The above mentioned form of uncertainty is expressed by the measure function μ in that the associated confidence intervals become larger:

$$\textbf{if } \delta \preceq \delta' \textbf{ then } \mu(\delta) \subseteq \mu(\delta')$$

In order to arrive at a proper and intuitive model-theoretic semantics which mirrors intuitions such as **if** ϕ *is confirmed* ($C\phi$) **then** ϕ *is likely* ($L\phi$), we will focus here on *well-behaved* frames $\mathcal{F}$ which enforce the existence of edges in $\mathcal{W}$, given $\preceq$ and $\delta, \delta^{\uparrow} \in \triangle$:

$$\textbf{if } (w, u) \in \delta \textbf{ and } \delta \preceq \delta^{\uparrow} \textbf{ then } (w, u) \in \delta^{\uparrow}$$

However, by imposing this constraint, we also need to adapt the last case of the satisfiability relation:

5. **for all** $\delta \in \triangle$: $\mathcal{M}, w \models \delta\pi$ **iff** $\frac{\#\{u \mid (w,u) \in \delta^{\uparrow}, \delta \preceq \delta^{\uparrow}, \textbf{ and } \mathcal{M}, u \models \pi\}}{\#\{u \mid (w,u) \in \delta' \textbf{ and } \delta' \in \triangle\}} \in \mu(\delta)$

Not only are we scanning for edges (w, u) labeled with δ and for successor states u of w in which π holds in the denominator (original definition), but also take into account edges marked with more general modals $\delta^{\uparrow}$, s.t. $\delta^{\uparrow} \succeq \delta$. This mechanism implements a kind of *built-in model completion* that is not necessary in ordinary modal logics as they deal with only a *single* relation (viz., unlabeled arcs) that connects elements from $\mathcal{W}$ and the two modals $\Diamond$ and $\Box$ are defined in the usual dual way: $\Box\phi \equiv \neg\Diamond\neg\phi$.

5 Entailment Rules

This section addresses a restricted subset of entailment rules which will unveil new (or implicit) knowledge from graded medical statements. Recall that these kind of statements (in negation normal form) are a consequence of the application of simplification rules as depicted in Section 4.1. Thus, we assume a *preprocessing step* here that "massages" more complex statements that arise from a representation of graded (medical) statements in *natural language*. The entailments which we will present in a moment can either be *directly* implemented in a tuple-based reasoner, such as *HFC*, or in triple-based engines (e.g., Jena, OWLIM) which need to *reify* the medical statements in order to be compliant with the RDF triple model.

5.1 Modal Entailments

The entailments presented in this section deal with *plausible* inference centered around modals $\delta, \delta' \in \triangle$, some of them partly addressed in [17] in a pure OWL setting. We use the implication sign $\rightarrow$ to depict the entailment rules

$$lhs \rightarrow rhs$$

which act as *completion* (or *materialization*) rules the way as described in, e.g., [8] and [21], and used in today's semantic repositories. We sometimes even use the bi-conditional $\leftrightarrow$ to address that the LHS and the RHS are semantically equivalent, but will indicate the direction that should be used in a practical setting. As before, we define $\pi ::= p(\boldsymbol{c}) \mid \neg p(\boldsymbol{c})$.

We furthermore assume that for every modal $\delta \in \triangle$, a *complement* modal δ^{C} and a *mirror* modal δ^{M} exist (see Section 4.1).

Lift

$$(\mathsf{L}) \quad \pi \leftrightarrow \top\pi$$

This rule interprets propositional statements as special modal formulae. It might be dropped and can be seen as a pre-processing step. We have used it in the Hepatitis example above. Usage: left-to-right direction.

Generalize

$$(\mathsf{G}) \quad \delta\pi \wedge \delta \preceq \delta' \rightarrow \delta'\pi$$

This rule schema can be instantiated in various ways, using the modal hierarchy from Section 4.3; e.g., $\top\pi \rightarrow C\pi$, $C\pi \rightarrow L\pi$, or $E\pi \rightarrow U\pi$. It has been used in the Hepatitis example.

Complement

$$(\mathsf{C}) \quad \neg\delta\pi \leftrightarrow \delta^{\mathsf{C}}\pi$$

In principle, (C) is not needed in case the statement is already in negation normal form. This schema might be useful for natural language paraphrasing (explanation). Given $\triangle$, there are two possible instantiations, viz., $E\pi \leftrightarrow \neg N\pi$ and $N\pi \leftrightarrow \neg E\pi$ (note: $\mu(E) \cup \mu(N) = [0, 1]$).

Mirror

$$(\mathsf{M}) \quad \delta\neg\pi \leftrightarrow \delta^{\mathsf{M}}\pi$$

Again, (M) is in principle not needed as long as the modal proposition is in negation normal form, since we do allow for negated propositional statements $\neg p(\boldsymbol{c})$. This schema might be useful for natural language paraphrasing (explanation). For $\triangle$, there are six possible instantiations, viz., $E\pi \leftrightarrow C\neg\pi$, $C\pi \leftrightarrow E\neg\pi$, $L\pi \leftrightarrow U\neg\pi$, $U\pi \leftrightarrow L\neg\pi$, $\top\pi \leftrightarrow \bot\neg\pi$, and $\bot\pi \leftrightarrow \top\neg\pi$.

Uncertainty

$$(\mathsf{U}) \quad \delta\pi \wedge \neg\delta\pi \leftrightarrow \delta\pi \wedge \delta^{\mathsf{C}}\pi \leftrightarrow ?\pi$$

The *co-occurrence* of $\delta\pi$ and $\neg\delta\pi$ does *not* imply logical *inconsistency* (propositional case: $\pi \wedge \neg\pi$), but leads to complete *uncertainty* about the validity of π. Remember that $\mu(?) = \mu(\delta) \cup \mu(\delta^{\mathsf{C}}) = [0, 1]$ (usage: left-to-right direction):

$$\begin{array}{l} \quad\;\; 0 \qquad\qquad\qquad\quad 1 \\ \mu : |\!-\!\delta^{\mathsf{C}}\!-\!|\!-\!-\!-\!\delta\!-\!-\!-\!| \\ \qquad\;\; \pi \qquad\qquad \pi \end{array}$$

Negation

$$\text{(N)} \quad \delta(\pi \wedge \neg\pi) \leftrightarrow \delta\pi \wedge \delta\neg\pi \leftrightarrow \delta\pi \wedge \delta^{\mathsf{M}}\pi \leftrightarrow \delta^{\mathsf{M}}\neg\pi \wedge \delta^{\mathsf{M}}\pi \leftrightarrow \delta^{\mathsf{M}}(\pi \wedge \neg\pi)$$

(N) shows that $\delta(\pi \wedge \neg\pi)$ can be formulated equivalently using the mirror modal:

$$\begin{array}{ll} & \quad 0 \qquad\qquad\qquad\qquad 1 \\ \mu : & |\!\!-\!\delta^{\mathsf{M}}\!-\!|\!\!-\!\!-\!\!-\!\!-\!|\!-\!\delta\!-\!| \\ & \pi \wedge \neg\pi \qquad\quad \pi \wedge \neg\pi \end{array}$$

In general, (N) is *not* the modal counterpart of the *law of non-contradiction*, as $\pi \wedge \neg\pi$ is usually afflicted by vagueness, meaning that from $\delta(\pi \wedge \neg\pi)$, we can *not* infer that $\pi \wedge \neg\pi$ is the case for the concrete example in question (recall the intention behind the confidence intervals; see Section 3). There is one notable exception, involving the $\top$ and $\bot$ modals. This is formulated by the next entailment rule.

Error

$$\text{(E)} \quad \top(\pi \wedge \neg\pi) \leftrightarrow \bot(\pi \wedge \neg\pi) \rightarrow\, !(\pi \wedge \neg\pi)$$

(E) *is* the modal counterpart of the *law of non-contradiction* (recall: $\top = \bot^{\mathsf{M}}$ and $\bot = \top^{\mathsf{M}}$). For this reason and by definition, the *error* (or *failure*) modal ! from Section 3 comes into play here. The modal ! can serve as a hint to either stop a computation the first time it occurs or to continue reasoning, but to syntactically memorize the ground atoms (viz., π and $\neg\pi$) which have led to an inconsistency. Usage: left-to-right direction.

5.2 Subsumption Entailments

As before, we define two subsets of $\triangle$, called $\mathsf{1} = \{\top, C, L, N\}$ and $\mathsf{0} = \{\bot, E, U\}$, thus $\mathsf{1}$ and $\mathsf{0}$ effectively become

$$\mathsf{1} = \{\top, C, L, N, U^{\mathsf{C}}\} \qquad\qquad \mathsf{0} = \{\bot, U, E, C^{\mathsf{C}}, L^{\mathsf{C}}, N^{\mathsf{M}}\}$$

due to the use of complement modals δ^{C} and mirror modals δ^{M} for every base modal $\delta \in \triangle$ and by assuming that $E = N^{\mathsf{C}}$, $E = C^{\mathsf{M}}$, $U = L^{\mathsf{M}}$, and $\bot = \top^{\mathsf{M}}$, together with the four "opposite" cases.

Now let $\sqsubseteq$ abbreviate relation subsumption as known from description logics and realized in OWL through `rdfs:subClassOf` (class subsumption) and `rdfs:subPropertyOf` (property subsumption). Given these remarks, we define two further very practical and plausible modal entailments which can be seen as the modal extension of the entailment rules (rdfs9) (for classes) and (rdfs7) (for properties) in RDFS; see [8].

$$\text{(S1)} \quad \mathsf{1}p(\boldsymbol{c}) \wedge p \sqsubseteq q \rightarrow \mathsf{1}q(\boldsymbol{c}) \qquad\qquad \text{(S0)} \quad \mathsf{0}q(\boldsymbol{c}) \wedge p \sqsubseteq q \rightarrow \mathsf{0}p(\boldsymbol{c})$$

Note how the use of p and q switches in the antecedent and the consequent, even though $p \sqsubseteq q$ holds in both cases. Note further that propositional statements π are restricted to the positive case $p(\boldsymbol{c})$ and $q(\boldsymbol{c})$, as their negation in the antecedent will not lead to any valid entailments. Here are four *instantiations* of (S0) and (S1) (remember, $C \in \mathsf{1}$ and $E \in \mathsf{0}$):

$C\mathsf{ViralHepatitisB}(x) \wedge \mathsf{ViralHepatitisB} \sqsubseteq \mathsf{ViralHepatitis} \rightarrow C\mathsf{ViralHepatitis}(x)$
$E\mathsf{Hepatitis}(x) \wedge \mathsf{ViralHepatitis} \sqsubseteq \mathsf{Hepatitis} \rightarrow E\mathsf{ViralHepatitis}(x)$

$C\mathsf{deeplyEnclosedIn}(x, y) \wedge \mathsf{deeplyEnclosedIn} \sqsubseteq \mathsf{containedIn} \rightarrow C\mathsf{containedIn}(x, y)$
$E\mathsf{containedIn}(x, y) \wedge \mathsf{superficiallyLocatedIn} \sqsubseteq \mathsf{containedIn}$
$\rightarrow E\mathsf{superficiallyLocatedIn}(x, y)$

5.3 Extended RDFS & OWL Entailments

In this section, we will consider some of the entailment rules for RDFS [8] and a restricted subset of OWL [21]. Remember that modals only head literals π, neither TBox nor RBox axioms. Concerning the original entailment rules, we will distinguish *four principal cases* to which the extended rules belong (we will only consider the unary and binary case here as used in description logics/OWL):

1. TBox and RBox axiom schemas will not undergo a modal extension;
2. rules get extended in the antecedent;
3. rules take over the modal from the antecedent to the consequent;
4. rules aggregate several modals from the antecedent in the consequent.

We will illustrate the individual cases in the following subsections with examples by using a kind of description logic syntax. Clearly, the set of extended entailments depicted here is *not complete*.

Case-1 Rules: No Modals Entailment rule rdfs11 from [8] deals with class subsumption: $\mathsf{C} \sqsubseteq \mathsf{D} \wedge \mathsf{D} \sqsubseteq \mathsf{E} \rightarrow \mathsf{C} \sqsubseteq \mathsf{E}$. As this is a terminological axiom schema, the rule stays *constant* in the modal domain. Example:

$\mathsf{ViralHepatitisB} \sqsubseteq \mathsf{ViralHepatitis} \wedge \mathsf{ViralHepatitis} \sqsubseteq \mathsf{Hepatitis}$
$\rightarrow \mathsf{ViralHepatitisB} \sqsubseteq \mathsf{Hepatitis}$

Case-2 Rules: Modals on LHS, No or $\top$ Modals on RHS The following original rule rdfs3 from [8] imposes a range restriction on objects of binary ABox relation instances: $\forall \mathsf{P.C} \wedge \mathsf{P}(\mathsf{x}, \mathsf{y}) \rightarrow \mathsf{C}(\mathsf{y})$.
The extended version (which we call Mrdfs3) needs to address the proposition in the antecedent, but must not change the consequent (even though we always use the $\top$ modality here for typing; see Section 2):

$$(\mathsf{Mrdfs3}) \quad \forall \mathsf{P.C} \wedge \delta \mathsf{P}(\mathsf{x}, \mathsf{y}) \rightarrow \top \mathsf{C}(\mathsf{y})$$

Example: $\forall \mathsf{suffersFrom.Disease} \wedge L\mathsf{suffersFrom}(x, y) \rightarrow \top \mathsf{Disease}(y)$

Case-3 Rules: Keeping LHS Modals on RHS Inverse properties switch their arguments [21]: $\mathsf{P} \equiv \mathsf{Q}^{-} \wedge \mathsf{P}(\mathsf{x}, \mathsf{y}) \rightarrow \mathsf{Q}(\mathsf{y}, \mathsf{x})$.
The extended version of rdfp8 simply keeps the modal operator:

$$(\mathsf{Mrdfp8}) \quad \mathsf{P} \equiv \mathsf{Q}^{-} \wedge \delta \mathsf{P}(\mathsf{x}, \mathsf{y}) \rightarrow \delta \mathsf{Q}(\mathsf{y}, \mathsf{x})$$

Example: $\mathsf{containedIn} \equiv \mathsf{contains}^{-} \wedge C\mathsf{containedIn}(x, y) \rightarrow C\mathsf{contains}(y, x)$

Case-4 Rules: Aggregating LHS Modals on RHS Now comes the most interesting case of modalized RDFS/OWL entailment rules that offers several possibilities on a varying scale between *skeptical* and *credulous* entailments, depending on the degree of uncertainty, as expressed by the measuring function μ of the modal operator. Consider the original rule $\mathsf{rdfp4}$ from [21] for transitive properties P: $\mathsf{P}^+ \sqsubseteq \mathsf{P} \wedge \mathsf{P}(\mathsf{x},\mathsf{y}) \wedge \mathsf{P}(\mathsf{y},\mathsf{z}) \rightarrow \mathsf{P}(\mathsf{x},\mathsf{z})$.
How does the modal on the RHS of the extended rule look like, depending on the two LHS modals? There are several possibilities. By operating directly on the *modal hierarchy*, we are allowed to talk about, e.g., the *least upper bound* or the *greatest lower bound* of δ and δ'. When taking the associated *confidence intervals* into account, we might even play with the low and high number of the intervals, say, by applying the *arithmetic mean* or simply by *multiplying* the corresponding numbers.
Let us first consider the general rule from which more specialized versions can be derived, simply by instantiating the combination operator $\odot$:

$$(\mathsf{Mrdfp4}) \quad \mathsf{P}^+ \sqsubseteq \mathsf{P} \wedge \delta\mathsf{P}(\mathsf{x},\mathsf{y}) \wedge \delta'\mathsf{P}(\mathsf{y},\mathsf{z}) \rightarrow (\delta \odot \delta')\mathsf{P}(\mathsf{x},\mathsf{z})$$

Here is an instantiation of $\mathsf{Mrdfp4}$ dealing with the transitive relation $\mathsf{contains}$ from above: $C\mathsf{contains}(x,y) \wedge L\mathsf{contains}(y,z) \rightarrow (C \odot L)\mathsf{contains}(x,z)$

What is the result of $C \odot L$ here? It depends. Probably both on the application domain and the epistemic commitment one is willing to accept about the "meaning" of gradation words/modal operators. To enforce that $\odot$ is at least both *commutative* and *associative* is probably a good idea, making the sequence of modal clauses order-independent.

5.4 Custom Entailments

Custom entailments are inference rules that are not derived from universal non-modalized RDFS and OWL entailment rules (Section 5.3), but have been formulated to capture the domain knowledge of experts (e.g., physicians). Here is an example. Consider that Hepatitis B is an infectious disease

$\mathsf{ViralHepatitisB} \sqsubseteq \mathsf{InfectiousDisease} \sqsubseteq \mathsf{Disease}$

and note that there exist vaccines against it. Assume that the liver l of patient p quite hurts (modal C), but p has been definitely vaccinated (modal $\top$) against Hepatitis B before:

$C\mathsf{hasPain}(p,l) \wedge \top\mathsf{vaccinatedAgainst}(p,\mathsf{ViralHepatitisB})$

Given that p received a vaccination, the following custom rule will *not* fire (x and y below are now universally-quantified variables; z an existentially-quantified RHS-only variable):

$\top\mathsf{Patient}(x) \wedge \top\mathsf{Liver}(y) \wedge C\mathsf{hasPain}(x,y) \wedge U\mathsf{vaccinatedAgainst}(x,\mathsf{ViralHepatitisB})$
$\rightarrow N\mathsf{ViralHepatitisB}(z) \wedge N\mathsf{suffersFrom}(x,z)$

Now assume another person p' that is pretty sure (s)he was never vaccinated:

$E\mathsf{vaccinatedAgainst}(p',\mathsf{ViralHepatitisB})$

Given the above custom rule, we are allowed to infer that (h instantiation of z)

$N\mathsf{ViralHepatitisB}(h) \wedge N\mathsf{suffersFrom}(p', h)$

The subclass axiom from above thus assigns

$N\mathsf{InfectiousDisease}(h)$

so that we can query for patients for whom an infectious disease is *not unlikely*, in order to initiate appropriate methods (e.g., further medical investigations).

6 Related Approaches and Remarks

It is worth noting to state that this paper is interested in the representation of and reasoning with *uncertain assertional* knowledge, and neither in dealing with *vagueness* found in natural language (*very small*), nor in handling *defaults* and *exceptions* in *terminological* knowledge (*penguins can't fly*).

To the best of our knowledge, the modal logic presented in this paper uses for the first time modal operators for expressing the degree of (un)certainty of propositions. These modal operators are interpreted in the model theory through confidence intervals, by using a measure function μ. From a model point of view, our modal operators are related to *counting modalities* $\Diamond^{\geq k}$ [6, 1]—however, we do *not* require a *fixed* number $k \in \mathbb{N}$ of reachable successor states (*absolute* frequency), but instead *divide* the number of worlds v reached through label $\delta \in \triangle$ by the number of all reachable worlds, given current state w, yielding $0 \leq p \leq 1$. This fraction then is further constrained by requiring $p \in \mu(\delta)$ (*relative* frequency), as defined in case 5. of the satisfaction relation in Sections 4.2 and 4.3.

As [23] precisely put it: "*... what axioms and rules must be added to the propositional calculus to create a usable system of modal logic is a matter of philosophical opinion, often driven by the theorems one wishes to prove ...*". Clearly, the logic Λ is *no* exception and its design is driven by commonsense knowledge and plausible inferences, we try to capture.

Our modal logic can be regarded as an instance of the *normal* modal logic $\mathbf{K} := (N) + (K)$ when identifying the basic modal operator $\Box$ with the modal $\top$ (and *only* with $\top$) and by enforcing the *well-behaved* frame condition from Section 4.3. Given $\Box \equiv \top$, Λ then includes the *necessitation rule* (N) $p \rightarrow \top p$ and the *distribution axiom* (K) $\top(p \rightarrow q) \rightarrow (\top p \rightarrow \top q)$ where p, q being special theorems in Λ, viz., positive and negative propositional letters.

(N) can be seen as a special case of (L), the *Lift* modal entailment (left-to-right direction) from Section 5.1. (K) can be proven in Λ by choosing $\top \in \underline{1}$ in simplification rule *8b* (Section 4.1) and by instantiating (G), the *Generalize* modal entailment (Section 5.1), together with the application of the tautology $(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$:

$$\cfrac{\cfrac{\cfrac{\cfrac{\top(p \rightarrow q) \rightarrow (\top p \rightarrow \top q)}{\top(\neg p \vee q) \rightarrow (\neg\top p \vee \top q)}}{(\top\neg p \vee \top q) \rightarrow (\neg\top p \vee \top q)}}{\top\neg p \rightarrow \neg\top p}}{\underline{\bot p \rightarrow \top^{\mathsf{C}} p}}$$

The final simplification at which we arrive is valid, since $\bot \preceq \top^{\mathsf{C}}$:

$$\mu(\bot) = [0, 0] \subseteq [0, 1) = \mu(\top^{\mathsf{C}})$$

Again, through (L) (right-to-left direction), Λ also incorporates the *reflexivity axiom* (T) $\top p \rightarrow p$ making Λ (at least) an instance of the system **T**. However, this investigation is in a certain sense *useless* as it does *not* address the other modals: almost always, neither (N), (K), nor (T) hold for modals from $\triangle$. Thus, we can *not* view Λ as an instance of a *poly-modal* logic.

Several approaches to representing and reasoning with uncertainty have been investigated in *Artificial Intelligence* (see [16, 7] for two comprehensive overviews). Very less so has been researched in the *Description Logic* community, and little or nothing of this research has find its way into implemented systems. [9] and [10] consider *uncertainty* in $\mathcal{ALC}$ concept hierarchies, plus concept typing of individuals (unary relations) in different ways (probability values vs. intervals; conditional probabilities in TBox vs. ABox). They do not address uncertain binary (or even n-ary) relations. [22] investigates *vagueness* in $\mathcal{ALC}$ concept descriptions to address statements, such as *the patient's temperature is high*, but also for determining membership degree (*38.5 ℃*). This is achieved through *membership manipulators* which are functions, returning a truth value between 0 and 1, thus deviating from a two-valued logic. [20] defines a *fuzzy* extension of $\mathcal{ALC}$, based on Zadeh's *fuzzy logic*. As in [22], the truth value of an assertion is replaced by a membership value from $[0, 1]$. $\mathcal{ALC}$ assertions α in [20] are made fuzzy by writing, e.g., $\langle \alpha \geq n \rangle$, thus taking a single truth value from $[0, 1]$. An even more expressive description logic, Fuzzy OWL, based on OWL DL, is investigated in [19].

Our work might be viewed as a modalized version of a restricted fragment of *Subjective Logic* [11, 12], a probabilistic logic that can be seen as an extension of Dempster-Shafer belief theory. Subjective Logic addresses subjective believes by requiring numerical values for *believe* b, *disbelieve* d, and *uncertainty* u, called (subjective) opinions. For each proposition, it is required that $b + d + u = 1$. The translation from modals δ to $\langle b, d, u \rangle$ is determined by the length of the confidence interval $\mu(\delta) = [l, h]$ and its starting/ending numbers, viz., $u := h - l$, $b := l$, and $d := 1 - h$.

Acknowledgements

The research described in this paper has been co-funded by the Horizon 2020 Framework Programme of the European Union within the project PAL (Personal Assistant for healthy Lifestyle) under Grant agreement no. 643783. The authors have profited from discussions with our colleagues Miroslav Janíček and Bernd Kiefer and would like to thank the three reviewers for their suggestions.

References

1. Areces, C., Hoffmann, G., Denis, A.: Modal logics with counting. In: Proceedings of the 17th Workshop on Logic, Language, Information and Computation (WoLLIC). pp. 98–109 (2010)

2. Baader, F.: Description logic terminology. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.) The Description Logic Handbook, pp. 495–505. Cambridge University Press, Cambridge (2003)
3. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge (2001)
4. Davidson, D.: The logical form of action sentences. In: Rescher, N. (ed.) The Logic of Decision and Action, pp. 81–95. University of Pittsburgh Press (1967)
5. Devlin, K.: Situation theory and situation semantics. In: Gabbay, D.M., Woods, J. (eds.) Handbook of the History of Logic. Volume 7, pp. 601–664. Elsevier (2006)
6. Fine, K.: In so many possible worlds. Notre Dame Journal of Formal Logic 13(4), 516–520 (1972)
7. Halpern, J.Y.: Reasoning About Uncertainty. MIT Press, Cambridge, MA (2003)
8. Hayes, P.: RDF semantics. Tech. rep., W3C (2004)
9. Heinsohn, J.: ALCP – Ein hybrider Ansatz zur Modellierung von Unsicherheit in terminologischen Logiken. Ph.D. thesis, Univ. Saarlandes (Jun 1993), in German
10. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR). pp. 305–316 (1994)
11. Jøsang, A.: Artificial reasoning with subjective logic. In: Proceedings of the 2nd Australian Workshop on Commonsense Reasoning (1997)
12. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems 9(3), 279–311 (2001)
13. Krengel, U.: Einführung in die Wahrscheinlichkeitstheorie und Statistik. Vieweg, 7th edn. (2003), in German
14. Krieger, H.U.: A temporal extension of the Hayes/ter Horst entailment rules and an alternative to W3C's n-ary relations. In: Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS). pp. 323–336 (2012)
15. Krieger, H.U.: An efficient implementation of equivalence relations in OWL via rule and query rewriting. In: Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC). pp. 260–263 (2013)
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, CA, MA (1988)
17. Schulz, S., Martínez-Costa, C., Karlsson, D., Cornet, R., Brochhausen, M., Rector, A.: An ontological analysis of reference in health record statements. In: Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS 2014) (2014)
18. Sider, T.: Four Dimensionalism. An Ontology of Persistence and Time. Oxford University Press (2001)
19. Stoilos, G., Stamou, G.B., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Fuzzy OWL: uncertainty and the semantic web. In: Proceedings of the OWLED '05 Workshop on OWL: Experiences and Directions (2005)
20. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research 14, 147–176 (2001)
21. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3, 79–115 (2005)
22. Tresp, C.B., Molitor, R.: A description logic for vague knowledge. In: Proceedings of the 13th European Conference on Artificial Intelligence . pp. 361–365 (1998)
23. Wikipedia: Modal logic — Wikipedia, The Free Encyclopedia (2015), https://en.wikipedia.org/wiki/Modal_logic, [Online; accessed 19-June-2015]

Extending OWL Ontologies by Cartesian Types to Represent N-ary Relations in Natural Language

Hans-Ulrich Krieger & Christian Willms
German Research Center for Artificial Intelligence (DFKI)
krieger@dfki.de | c.willms1@gmx.de

Abstract

Arbitrary n-ary relations ($n \geq 1$) can in principle be realized through binary relations obtained by a reification process that introduces new individuals to which the additional arguments are linked via accessor properties. Modern ontologies which employ standards such as RDF and OWL have mostly obeyed this restriction, but have struggled with it nevertheless. Additional arguments for representing, e.g., valid time, grading, uncertainty, negation, trust, sentiment, or additional verb roles (for ditransitive verbs and adjuncts) are often better modeled in relation and information extraction systems as direct arguments of the relation instance, instead of being hidden in deep structures. In order to address non-binary relations directly, ontologies must be extended by *Cartesian* types, ultimately leading to an extension of the standard entailment rules for RDFS and OWL. In order to support ontology construction, ontology editors such as Protégé have to be adapted as well.

1 Decription Logics, OWL, and RDF

Relations in description logics (DLs) are either unary (so-called *concepts* or *classes*) or binary (*roles* or *properties*) predicates (Baader et al., 2003). As the designers of OWL (Smith et al., 2004; Hitzler et al., 2012) decided to be compatible with already existing standards, such as RDF (Cyganiak et al., 2014) and RDFS (Brickley and Guha, 2014), as well as with the universal RDF data object, the *triple*,

subject predicate object

a unary relation such as $C(a)$ (class membership) becomes a binary relation via the RDF `type` predicate:

```
a rdf:type C
```

For very good reasons (mostly for decidability), DLs usually restrict themselves to decidable function-free two-variable subsets of first-order predicate logic. Nevertheless, people have argued for relations of more than two arguments, some of them still retaining decidability and coming up with a better memory footprint and a better complexity for the various inference tasks than their triple-based relatives (Krieger, 2012). This idea conservatively extends the standard *triple-based* model towards a more general *tuple-based* approach ($n + 1$ being the arity of the *predicate*):

subject predicate object$_1 \ldots$ *object*$_n$

Using a standard relation-oriented notation, we often interchangeably write

$p(s, o_1, \ldots, o_n)$

Here is an example, dealing with *diachronic* relations (Sider, 2001), relation instances whose object values might change over time, but whose subject values coincide with each other. For example (quintuple representation),

```
peter marriedTo liz 1997 1999        peter marriedTo lisa 2000 2010
```

or (relation notation)

marriedTo(*peter, liz, 1997, 1999*)   *marriedTo*(*peter, lisa, 2000, 2010*)

which we interpret as the (time-dependent) statement that *Peter* was married to *Liz* from 1997 until 1999 and to *Lisa* from 2000–2010.

In a triple-based setting, semantically representing the same information requires a lot more effort. There already exist several approaches to achieve this (Krieger, 2014), all coming up with at least one brand-new individual (introduced by a hidden existential quantification), acting as an *anchor* to which the object information (the range information of the relation) is bound through additional properties (a kind of *reification*). For instance, the so-called *N-ary relation encoding* (Hayes and Welty, 2006), a W3C best-practice recommendation, sticks to binary relations/triples and uses a *container* object to encode the range information (`ppt1` and `ppt2` being the new individuals):

```
peter marriedTo ppt1
ppt1 rdf:type nary:PersonPlusTime
ppt1 nary:value liz
ppt1 nary:starts "1997"^^xsd:gYear
ppt1 nary:ends "1999"^^xsd:gYear
```

```
peter marriedTo ppt2
ppt2 rdf:type nary:PersonPlusTime
ppt2 nary:value lisa
ppt2 nary:starts "2000"^^xsd:gYear
ppt2 nary:ends "2010"^^xsd:gYear
```

As we see from this small example, a quintuple is represented by five triples. The relation name is retained, however, the range of the relation changes from, say, `Person` to the type of the container object which we call here `PersonPlusTime`.

Rewriting ontologies to the *latter* representation is clearly time consuming, as it requires further classes, redefines property signatures, and rewrites relation instances, as shown by the *marriedTo* example. In addition, reasoning and querying with such representations is extremely complex, expensive, and error-prone. Unfortunately, the *former* tuple-based representation which argues for additional (temporal) arguments is *not* supported by ontology editors today, as it would require to deal with general relations.

2 What this Paper is (Not) About & Related Approaches

We would like to make clear that this paper is *not* about developing a theory for yet another new DL which permits n-ary relations. The approach presented here suggests that the concepts of *domain* and *range* of a relation are still useful when extending a binary relation with more arguments, instead of talking about the *arity* of a relation in general. We furthermore suggest in Section 6 to introduce so-called *extra arguments* which neither belong to the domain nor the range of a relation, and can be seen, as well, should be used as a kind of relation instance annotation. In the course of the paper, we also indicate that most of the entailment rules for RDFS (Hayes, 2004) and OWL Horst/OWL 2 RL (ter Horst, 2005; Motik et al., 2012) can be extended by Cartesian types and n-ary relations, and present an incomplete set of rules in Figure 1. Our approach takes a liberal stance in that it neither ask for the "nature" or "use" of the arguments (e.g., whether they are time points), nor for a (sound, complete, terminating, . . .) set of tableau or entailment rules. In fact, if we would take this into account, we would end up in a potentially infinite number of different sets of rules, some of them requiring additional (lightweight) tests and actions, going beyond simple symbol matching; see (Krieger, 2012) for such a set of rules that model valid time, turning binary relations into quaternary ones. For various reasons, we propose a general restriction on the use of Cartesian types in Section 5, viz., to avoid typing individuals with Cartesian types and to maintain still singleton typing. The practical accomplishment of this paper lies in an extension of the Protégé editor for Cartesian types and n-ary relations that should be complemented by application-independent, but also domain-specific rules for a given application domain (e.g., to address *valid time*).

Since the early days of KL-ONE, DLs supporting relations with more than two arguments have been discussed, e.g., NARY[KANDOR] (Schmolze, 1989), $\mathcal{CIFR}$ (De Giacomo and Lenzerini, 1994), $\mathcal{DLR}$ (Calvanese et al., 1997), or $\mathcal{GF}1^-$ (Lutz et al., 1999). Especially Schmolze (1989) argued that "the advantages for allowing *direct* representation of n-ary relations far outweigh the reasons for the restriction" (i.e., restricting $n \leq 2$). To the best of our knowledge and with the exception of NARY[KANDOR], these DL languages have still remained theoretical work. In (Krieger, 2013), we presented an implemented theory-agnostic forward chainer, called *HFC*, which is comparable to popular semantic repositories such as Jena or OWLIM and which supports arbitrary n-tuples. The engine is able to run non-builtin entailment rule sets à la OWL Horst/OWL 2 RL and comes with a conservative extension of these OWL dialects for *valid time* (Krieger, 2012). Further rule regimes are possible as long as they are expressible in *HFC*'s rule language which permits standard symbol matching, additional LHS tests, and RHS actions.

3 Extending Ontologies through Cartesian Types

Modern ontologies make use of standards, defined and coordinated by the W3C, such as XSD, RDF, RDFS, or OWL. OWL, as an instance of the description logic family, describes a domain in terms of classes (concepts), binary properties (roles), and instances (individuals). Complex expressions, so-called *axioms*, are defined via concept-forming operators (viz., subsumption and equivalence). The entirety of all such axioms which can be separated into those dealing with terminological knowledge (TBox), relational knowledge (RBox), and assertional knowledge (ABox), is usually called an *ontology* today.

Ontology editors which are geared towards RDF and OWL are thus *not* able to define n-ary relations *directly* in the RBox, nor are they capable of stating arbitrary tuples (instances of n-ary relations) in the ABox (together with Cartesian types in the TBox; see below). This would require an extension of the triple data model, or equivalently, allowing for n-ary relations ($n > 2$).

Formally, the extension of a binary relation p, can be seen as a (potentially infinite) set of pairs (s, o), coming from the Cartesian product of its *domain* $\mathbb{D}$ and *range* $\mathbb{R}$: $p \subseteq \mathbb{D} \times \mathbb{R}$. We then often say that a relation p is *defined* on $\mathbb{D}$, say, the *marriedTo* relation is *defined* on *Person*.

Now, in order to allow for more than two arguments, we decompose $\mathbb{R}$, leading to $p \subseteq \mathbb{D} \times \mathbb{R}_1 \times \cdots \times \mathbb{R}_n$. Note that we still make a distinction between domain $\mathbb{D}$ and range $\mathbb{R} = \mathbb{R}_1 \times \cdots \times \mathbb{R}_n$, and still say that p is *defined* on $\mathbb{D}$. Coming back to the previous section and the quaternary *marriedTo* relation, we can say that

$$\mathit{marriedTo} \subseteq \mathit{Person} \times \mathit{Person} \times \mathit{Year} \times \mathit{Year}$$

For reasons that will become clear in a moment, not only the range but also the domain of a relation can, in principle, be deconstructed: $p \subseteq (\mathbb{D}_1 \times \cdots \times \mathbb{D}_m) \times (\mathbb{R}_1 \times \cdots \times \mathbb{R}_n)$. When it is clear from the context, we often omit the parentheses and simply write $p \subseteq \mathbb{D}_1 \times \cdots \times \mathbb{D}_m \times \mathbb{R}_1 \times \cdots \times \mathbb{R}_n$. We then say that the domain of p is $\mathbb{D}_1 \times \cdots \mathbb{D}_m$ and the range is $\mathbb{R}_1 \times \cdots \times \mathbb{R}_n$, thus p becomes an $(m + n)$-ary relation. Again, we say that p is defined on $\mathbb{D}_1 \times \cdots \mathbb{D}_m$.

Graphically, such an extension is easy write down. Let us start, again, with binary relations and let us picture the resulting graph for the following set $\{p(a, b), q(b, c), q(b, d), r(b, e)\}$ of binary relation instances by using directed labeled edges:

$$\begin{array}{ccccc} & & & & c \\ & & & {\scriptstyle q}\nearrow & \\ a & \xrightarrow{p} & b & \xrightarrow{q} & d \\ & & & {\scriptstyle r}\searrow & \\ & & & & e \end{array}$$

Ontology editors such as Protégé (Horridge, 2004) essentially use such a representation: properties are defined on certain classes and ontology or ABox) population reduces to filling missing range arguments for specific instances.

But how do we depict the following set of relation instances

$$\{r((a, b, c), (d)), p((a, b, c), (a, x)), q((a, x), (y, z)\}$$

of arity 4 and 5, resp? Quite easy, simply by replacing individuals (= singles) in domain and range position through general tuples:

$$(d) \xleftarrow{r} (a, b, c) \xrightarrow{p} (a, x) \xrightarrow{q} (y, z)$$

The "problem" with this kind of graph representation is that we are still using a kind of container (denoted by the parentheses) which groups both domain elements $\mathbb{D}_i$ ($1 \leq i \leq m$) and range elements $\mathbb{R}_j$ ($1 \leq j \leq n$). But this is something we want to avoid as explicated before (recall the *N-ary relation encoding* example from Section 1).

The answer to all this is already laying before us and has already been introduced, viz., *Cartesian* types (remember the $\times_i \mathbb{D}_i$ and $\times_j \mathbb{R}_j$ notation). This, however, will require to extend the descriptive expressiveness of the TBox, RBox, and ABox of an ontology.

4 Cartesian Types in TBox, RBox, and ABox

Protégé (and other ontology editors such as TopBraid) displays the *class subsumption hierarchy* using *indentation*, e.g.,

```
▽ Entity
  ▽ Object
    ▽ Agent
      ▷ Group
      ▽ Person
        ▷ Man
        ▷ Woman
```

These concepts can be seen as *singles* (or singletons), representing a Cartesian product of only one element. Thus the class `Person` can be seen as the tuple `(Person)`, consisting of one tuple element. Similarly, when considering the *marriedTo* relation, we might view the range type as the Cartesian type `(Person,Year,Year)`. Clearly, neither does `(Person)` subsume `(Person,Year,Year)`, nor does the opposite case hold—they are incompatible, for which we write

$$(\mathtt{Person}) \bowtie (\mathtt{Person}, \mathtt{Year}, \mathtt{Year})$$

However, the following subsumption relations do hold, given the above type hierarchy:

$$(\mathtt{Man}, \mathtt{Year}, \mathtt{Year}) \sqsubseteq (\mathtt{Person}, \mathtt{Year}, \mathtt{Year})$$
$$(\mathtt{Woman}, \mathtt{Year}, \mathtt{Year}) \sqsubseteq (\mathtt{Person}, \mathtt{Year}, \mathtt{Year})$$
$$(\mathtt{Person}, \mathtt{Year}, \mathtt{Year}) \sqsubseteq (\mathtt{Agent}, \mathtt{Year}, \mathtt{Year})$$
$$(\mathtt{Group}, \mathtt{Year}, \mathtt{Year}) \sqsubseteq (\mathtt{Agent}, \mathtt{Year}, \mathtt{Year})$$

Now let $\mathcal{C}$ denote the set of concepts, $\mathcal{R}$ denote the set of all relations, and $\mathcal{I}$ denote the set of all instances. Quite naturally, the subsumption relation for concepts $\sqsubseteq \subseteq \mathcal{C} \times \mathcal{C}$ can be easily extended to Cartesian types:

$$\times_{i=1}^{m} C_i \sqsubseteq \times_{j=1}^{n} D_i \textbf{ iff } m = n \textbf{ and } C_i \sqsubseteq D_i, \textbf{ for all } i \in \{1, \ldots, m\}$$

Given such an extension, many of the standard entailment rules from (Hayes, 2004) and (ter Horst, 2005) can be easily adjusted, but also two new rules, called (ctsub) and (ctequiv), need to be introduced which propagate Cartesian type subsumption and equivalence down to their component classes (see Figure 1 for a representative, non-complete set of extended rules).

5 A Restriction on the Use of Cartesian Types

The extension introduced so far would even allow us to type *individuals* $a \in \mathcal{I}$ with any Cartesian type $\times_{i=1}^{m} C_i$ ($m \geq 1$) for which we might then write $\times_{i=1}^{m} C_i(a)$. This would make it possible to naturally extend, e.g., the universal instantiation schema (rdfs9) from Hayes (2004) with Cartesian types, viz.,

$$\text{(rdfs9)} \;\; \times_{i=1}^{m} C_i(a) \wedge \times_{i=1}^{m} C_i \sqsubseteq \times_{i=1}^{m} D_i \rightarrow \times_{i=1}^{m} D_i(a)$$

Such an extension is attractive, but has severe drawbacks. It makes domain and range inference more complex and would require a stronger descriptive apparatus, as it will become necessary to group and access *parts* of the domain and/or range arguments in order to indicate the true number of arguments of a relation, but also to indicate the proper argument types. This would become important when checking relation instances against their relation signature.

Consider, for instance, a quaternary relation $p \subseteq \mathbb{D} \times \mathbb{R}_1 \times \mathbb{R}_2 \times \mathbb{R}_3$ that seems to come with three *range* arguments. However, by typing individuals with Cartesian types, the above relation can be binary, ternary (two possibilities), or quaternary, depending on how we interpret the range arguments:

- $p \subseteq \mathbb{D} \times (\mathbb{R}_1 \times \mathbb{R}_2 \times \mathbb{R}_3)$
- $p \subseteq \mathbb{D} \times \mathbb{R}_1 \times \mathbb{R}_2 \times \mathbb{R}_3$
- $p \subseteq \mathbb{D} \times (\mathbb{R}_1 \times \mathbb{R}_2) \times \mathbb{R}_3$
- $p \subseteq \mathbb{D} \times \mathbb{R}_1 \times (\mathbb{R}_2 \times \mathbb{R}_3)$

And there are even further complex embeddings possible (remember type theory), such as

- $p \subseteq \mathbb{D} \times (\mathbb{R}_1 \times (\mathbb{R}_2 \times \mathbb{R}_3))$
- $p \subseteq \mathbb{D} \times ((\mathbb{R}_1 \times \mathbb{R}_2) \times \mathbb{R}_3)$

(ctsub) $\times_{i=1}^{m} C_i \sqsubseteq \times_{i=1}^{m} D_i \rightarrow \bigwedge_{i=1}^{m} C_i \sqsubseteq D_i$

(rdfs11) $\times_{i=1}^{m} C_i \sqsubseteq \times_{i=1}^{m} D_i \wedge \times_{i=1}^{m} D_i \sqsubseteq \times_{i=1}^{m} E_i \rightarrow \times_{i=1}^{m} C_i \sqsubseteq \times_{i=1}^{m} E_i$

(ctequiv) $\times_{i=1}^{m} C_i \equiv \times_{i=1}^{m} D_i \rightarrow \bigwedge_{i=1}^{m} C_i \equiv D_i$

(rdfp12c) $\times_{i=1}^{m} C_i \sqsubseteq \times_{i=1}^{m} D_i \wedge \times_{i=1}^{m} D_i \sqsubseteq \times_{i=1}^{m} C_i \rightarrow \times_{i=1}^{m} C_i \equiv \times_{i=1}^{m} D_i$

(rdfs2) $\forall P^{-}. \times_{i=1}^{m} C_i \wedge P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j) \rightarrow \bigwedge_{i=1}^{m} C_i(a_i)$

(rdfs3) $\forall P. \times_{j=1}^{n} D_j \wedge P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j) \rightarrow \bigwedge_{j=1}^{n} D_j(b_j)$

(rdfs7x) $P \sqsubseteq Q \wedge P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j) \rightarrow Q(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j)$

(rdfp1) $\leq 1P \wedge P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j) \wedge P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} c_j) \rightarrow \bigwedge_{j=1}^{n} \{b_j\} \equiv \{c_j\}$

(rdfp3) $P \equiv P^{-} \wedge P(\times_{i=1}^{m} a_i, \times_{i=1}^{m} b_i) \rightarrow P(\times_{i=1}^{m} b_i, \times_{i=1}^{m} a_i)$

(rdfp4) $P^{+} \sqsubseteq P \wedge P(\times_{i=1}^{m} a_i, \times_{i=1}^{m} b_i) \wedge P(\times_{i=1}^{m} b_i, \times_{i=1}^{m} c_i) \wedge \rightarrow P(\times_{i=1}^{m} a_i, \times_{i=1}^{m} c_i)$

Figure 1: Entailment rules using Cartesian types ($C_i, D_j, E_k \in \mathcal{C}$; $P, Q \in \mathcal{R}$; $a., b., c. \in \mathcal{I}$). Note that the notation $P(\times_{i=1}^{m} a_i, \times_{j=1}^{n} b_j)$ in the above rules does *not* indicate that P is a binary relation, but instead is of arity $m + n$ *and* $a_1, \ldots, a_m$ are the domain and $b_1, \ldots, b_n$ the range arguments for this specific relation instance of P. The names for the extended rule schemata are taken from (Hayes, 2004) and (ter Horst, 2005). (ctsub) and (ctequiv) are brand-new entailment rules for Cartesian types. The correctness of (rdfp4), addressing the transitivity of P, depends on the interpretation of the application domain (for instance, whether certain arguments are employed for expressing the validity of a fluent (the atemporal fact) over time; see also Section 6).

Mainly for this reason, we enforce that atomic *individuals* from $\mathcal{I}$ can only be typed to *single* concepts (singletons), and thus the relation signature

$$p \subseteq \mathbb{D}_1 \times \cdots \times \mathbb{D}_m \times \mathbb{R}_1 \times \cdots \times \mathbb{R}_n$$

is intended to mean that p takes *exactly* m domain arguments and *exactly* n range arguments, such that $\mathbb{D}_1, \ldots, \mathbb{D}_m, \mathbb{R}_1, \ldots, \mathbb{R}_n \in \mathcal{C}$ must be the case.

6 Extra Arguments

This section deals with what we call *extra arguments*, arguments that neither belong to the domain nor the range of an $(m + n)$-ary relation, but can be seen as a kind of additional *annotation*, belonging to specific relation instances.[1]

Let us start with a binary relation $(m, n = 1)$ and consider, again, the non-temporal version of *marriedTo* which is a true *symmetric* relation, expressed by the following instantiated entailment rule:

$$\textit{marriedTo}(i, j) \rightarrow \textit{marriedTo}(j, i)$$

Now, if we add time (b = begin; e = end), it becomes a quaternary relation as indicated before (for better readability, we separate the domain and range arguments from one another by using parentheses):

$$\checkmark\ \textit{marriedTo}(i, (j, b, e)) \rightarrow \textit{marriedTo}(j, (i, b, e))$$

In this sense, the temporal interval $[b, e]$ specifies the valid time in which the fluent (the atemporal statement) *marriedTo*(i, j) is true. By applying the extended rule (rdfp3) from Figure 1 for symmetry, we see that something clearly goes wrong:

$$\lightning\ \textit{marriedTo}(i, (j, b, e)) \rightarrow \textit{marriedTo}((j, b, e), i)$$

[1]This is like having annotation properties for *relation instances*, but OWL unfortunately offers this service only for classes, properties, and individuals.

as symmetric relations assume the same number of arguments in domain and range position! Our example above thus needs to be modified. One solution would be to reduplicate the starting and ending points, so we would end up in a sexternary relation:

$$\textit{marriedTo}((i, \underline{b, e}), (j, \underline{b, e})) \rightarrow \textit{marriedTo}((j, \underline{b, e}), (i, \underline{b, e}))$$

This is *not* an appealing solution as the structures become larger, and rules and queries are harder to formulate, read, debug, and process. We thus like to extend relations $p \subseteq \mathbb{D}_1 \times \cdots \times \mathbb{D}_m \times \mathbb{R}_1 \times \cdots \times \mathbb{R}_n$ by further arguments $\mathbb{A}_1 \times \cdots \times \mathbb{A}_o$, so that p becomes

$$p \subseteq \mathbb{D}_1 \times \cdots \times \mathbb{D}_m \times \mathbb{R}_1 \times \cdots \times \mathbb{R}_n \times \mathbb{A}_1 \times \cdots \times \mathbb{A}_o$$

or simply write $p \subseteq \mathbb{D} \times \mathbb{R} \times \mathbb{A}$. For the *marriedTo* example, we might choose Person from the ontology above and the XSD type gYear: $\mathbb{D}$ = Person, $\mathbb{R}$ = Person, $\mathbb{A}$ = gYear $\times$ gYear.

Thus by having these *extra* arguments, we can keep the entailment rules from Figure 1, extended, of course, by the additional annotations.[2] Besides having extra arguments for *valid time*, other areas are conceivable here, viz., *transaction time, space, sentiment, uncertainty, negation, vagueness, or graded information*.

7 Extensions to Protégé

In order to make Cartesian types available in Protégé, we will extend the *OWL Classes*, *Properties*, and *Individuals* tabs.

TBox: *OWL Classes* Tab

- *subclass explorer* pane (left column)
  extension of the subclass hierarchy towards Cartesian types.
- *class editor* pane (right column)
  depicting the right properties defined on a Cartesian type (domain); depicting the right Cartesian range types for the defined properties.

RBox: *Properties* Tab

- *property browser* pane (left column)
  extension of the property hierarchy towards Cartesian types.
- *property editor* pane (right column)
  extension of the domain and range boxes towards Cartesian types.
- **new:** *extra arguments* (part of the *property editor* pane)
  further definition box for the extra arguments.

ABox: *Individuals* Tab

- *class browser* pane (left column)
  extension of the subclass hierarchy towards Cartesian types.
- *instance browser* pane (middle column)
  possibility to generate sequence instances defined on Cartesian types (= sequences of instances of singleton types).
- *property editor* pane (right column)
  depicting the right properties defined on a sequence instance; allowing to choose or construct the range arguments; allowing to choose or construct the extra arguments.

Not only the graphical user interface needs to be extended, but also the internal representation (representation of tuples instead of triples), together with a modification of the input and output routines. We plan to have finished a first version of the extensions to Protégé in Spring 2015 and to present it at the workshop.

[2] Depending on the application domain, these annotations might find their way as (potentially aggregated) extra arguments in the relation instances of the consequence of a rule, e.g., in (rdfp4). We will look into this in more detail at the workshop.

Acknowledgements

The research described in this paper has been partially financed by the European project PAL (Personal Assistant for healthy Lifestyle) under Grant agreement no. 643783-RIA Horizon 2020. The authors have profited from discussions with our colleague Bernd Kiefer and would like to thank the three reviewers for their suggestions.

References

Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (2003). *The Description Logic Handbook*. Cambridge: Cambridge University Press.

Brickley, D. and R. Guha (2014). RDF Schema 1.1. Technical report, W3C.

Calvanese, D., G. De Giacomo, and M. Lenzerini (1997). Conjunctive query containment in description logics with n-ary relations. In *Proceedings of the International Workshop on Description Logics*, pp. 5–9.

Cyganiak, R., D. Wood, and M. Lanthaler (2014). RDF 1.1 concepts and abstract syntax. Technical report, W3C.

De Giacomo, G. and M. Lenzerini (1994). Description logics with inverse roles, functional restrictions, n-ary relations. In *Proceedings of the 4th European Workshop on Logics in Artificial Intelligence*, pp. 332–346.

Hayes, P. (2004). RDF semantics. Technical report, W3C.

Hayes, P. and C. Welty (2006). Defining N-ary relations on the Semantic Web. Technical report, W3C.

Hitzler, P., M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph (2012). OWL 2 web ontology language primer (second edition). Technical report, W3C.

Horridge, M. (2004). A practical guide to building OWL ontologies with the Protégé-OWL plugin. Technical report, University of Manchester.

Krieger, H.-U. (2012). A temporal extension of the Hayes/ter Horst entailment rules and an alternative to W3C's n-ary relations. In *Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS)*, pp. 323–336.

Krieger, H.-U. (2013). An efficient implementation of equivalence relations in OWL via rule and query rewriting. In *Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC)*, pp. 260–263.

Krieger, H.-U. (2014). A detailed comparison of seven approaches for the annotation of time-dependent factual knowledge in RDF and OWL. In *Proceedings of the 10th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*.

Lutz, C., U. Sattler, and S. Tobies (1999). A suggestion for an n-ary description logic. In *Proceedings of the International Workshop on Description Logics*, pp. 81–85.

Motik, B., B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz (2012). OWL 2 web ontology language profiles. Technical report, W3C.

Schmolze, J. G. (1989). Terminological knowledge representation systems supporting n-ary terms. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pp. 432–443.

Sider, T. (2001). *Four Dimensionalism. An Ontology of Persistence and Time*. Oxford University Press.

Smith, M. K., C. Welty, and D. L. McGuinness (2004). OWL Web Ontology Language Guide. Technical report, W3C.

ter Horst, H. J. (2005). Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *Journal of Web Semantics 3*, 79–115.

B Technical Reports

OpenCCG and LF Grammars for PAL

Stefania Racioppa, DFKI GmbH

January 26, 2016

Abstract

In the past year, we developed a large Italian OpenCCG grammar [1] [4] covering all utterances of the Italian end English *canned text*. In the current version, this grammar generates for the parsed utterances the same semantic output as the English OpenCCG grammar Moloko [3], which was used in several DFKI projects like CogX and NIFTi.

For Italian, we also developed a LF (logical form) grammar, integrating the semantic output of the Italian OpenCCG grammar in the Content planner [5]: this version supports all the Speech acts and variants defined in the current *canned text* version, but in the output side the *strings* and *string parts* were substituted by the *logical forms* parsed by the OpenCCG grammar. This makes the grammar maintenance and the realizer parameterization more comfortable. The string input for the TTS module is then generated by the OpenCCG surface realizer.

Besides this, the OpenCCG grammar was improved in terms of coverage and parsing precision, the verbalization for the interaction with groups was implemented in the Content planner grammars, and the questions and answers *strings* from the Quiz game database were fully integrated in the LF grammar.

1 Overview

The Content planner [5] is responsible in **PAL** for the modeling of the verbal output. This output is defined currently in a *canned text*, which consists of a large range of activity-related Speech acts in *string* form. Every output may consist of a single or more sentences, which can be concatenated or used alternatively:

```
:dvp ^ <SpeechAct>greeting
^ <Context>(<Familiarity>(yes|no) ^ <Encounter>notfirst ^ <Expressive>no)
->
###greet = random("ciao ", "buongiorno ", "ehi, ciao "),
###polite = ", come stai ",

###var1 = :canned ^ <stringOutput>concatenate(###greet,###polite)
          ^ <SpeechModus>interrogative,
###var2 = :canned ^ <stringOutput>###greet ^ <SpeechModus>indicative,
# ^ random(###var1,###var2):.
```

However, planning a natural verbal output using simple strings can become uncomfortable in some cases: e.g. in Italian the adjective inflection is given by the child (or the *activity* name) gender; the type of some prepositions depends on the following word (*conosci la risposta* ***alla prima*** *domanda* vs. ***a questa*** *domanda*). Again, some of the canned text variants affect the word order of the planned utterance. Each synctactical variant requires different strings and consequently *different outputs* in the canned text, which makes maintaining the rule consistency more difficult.

On the other hand, *logical forms* (LF) like the semantic output of the OpenCCG grammar represent the found words and sentences using only their canonical form, while attibutes like gender or number are outputted as feature, not as string. As features can be parameterized more easily than strings, i.e. just using simple variables, substituting the output strings of the canned text with logical forms makes the rule maintenance more comfortable.

Using the OpenCCG surface realizer, we then can build from the defined LF the surface form of the required utterance, which again is the input for the TTS system.

2 OpenCCG grammar

> OpenCCG is an open source natural language processing library written in Java, which provides parsing and realization services based on Mark Steedman's Combinatory Categorial Grammar (CCG) formalism [6]. The library makes use of the multi-modal extensions to CCG devised by Jason Baldridge in his dissertation [1] and in a joint EACL-03 paper with Geert-Jan Kruijff [2].

This brief description from [4] resumes the main features of the system employed for the development of the Italian CCG grammar for PAL. In the following we will describe the lexical families and categories defined in this phase. The Italian grammar uses the same semantics as in the English grammar MOLOKO, which is being developed at DFKI and used in several other projects, e.g., CogX and NIFTi. Using the same semantics facilitates the resources reusability.

2.1 Structure of the Grammar

A standard CCG grammar usually consists of four main sections:

- In the **type hierarchy**, we define the syntactical, morphological and semantical features to be used in the morphology and lexicon, e.g. (`GEN<15,10>: fem+masc {fem masc};`). The defined features are propagated to the category elements using IDs (e.g. `<15,10>`). Semantics can be declared in an appropriate ontology, if needed.
- In the **morphology**, we specify the words to be recognized by the grammar and the syntactical features related to them. Declaring a part of speech for each word, we associate it to a specific *family* (e.g. `tu: Pron(person): 2nd s-sg sg nom perso;`). Each family can thus include more words, which serve as *reference* for the arguments we might define in the dedicated lexical definition. Words can also be assigned to more than one family, e.g. if a context-dependent definition is required.
- In the **lexicon**, we define the syntactical structures (*family*) in which the words can be used. For each family we might specify one or more entries, i.e. *lexical categories*, like `family Pron {entry: np<15>[T stem=*]: T (*);}`.

  Every entry is a list, enumerating the *result* category (the first element of a list) and its arguments [4]; however, a category might also consist of a single element. If arguments are defined, we associate each of them with a slash; its direction shows whether the argument is on the left (`\`) or on the right (`/`) of the *reference* word (see above).

  The most categories defined in this grammar cover PAL domain-specific constructions – i.e. feedback expressions, question and answer enumeration.
- We used the CCG **rules** section mainly to simplify the grammar by automating some constantly recurring structures, like Italian pro-drop (see 2.5.1); a dedicated rule block was developed to extract embedded pronouns from the imperative and infinitive compound verb forms like *facciamolo di nuovo* (imperative *facciamo* + pronoun *lo*) or *vuoi farlo di nuovo?* (infinitive *fare* + pronoun *lo*).

The main focus of this report lies on the definition of lexical categories. Please refer to [4] for more information about the OpenCCG system.

2.2 Type hierarchy

In the Italian grammar, we included the type definitions and the ontology used in the MOLOKO grammar, except for some particular language-specific definitions, e.g. `AGREE` for the agreement between determiners and nouns.

MOLOKO includes a richly sorted ontological type hierarchy. Its main peculiarity is the distinction between three broad categories: **entities** (e.g. concrete objects or things), **events** (e.g. dynamic processes, states) and **modifiers**, which can be applied to any of the three basic categories (see [3]).

2.2.1 Semantical and synctactical features

In MOLOKO, a further important distinction is made between semantical and synctactical features.

Syntactical features can be used as syntax categories inside the grammar, while semantical ones can only be used as *macro arguments* in dictionary entries [3]. Both semantical and synctactical features are sorted by the main semantic categories used in the grammar (**entities**, **events**, **modifier**).

Semantical features refer only to the category they are specified for (e.g. `<T:Num>` for **entities**, as they usually become the semantic identifier `[T]` in the lexicon). These features automatically generate additional output nodes, if an entry containing them is parsed.

```
NUMBER<T:Num>: sg pl;
POLARITY<E:Polarity>: pos neg;
```

Synctactical features are propagated to the corresponding lexical category using IDs (e.g. `<10>`) and thus can be used by *every* category element marked with an allowed ID. Unlike the semantical features, synctactical features are used in the lexicon to *select* the dependencies of a lexical category or to enforce the agreement between compatible category elements.

```
GEN<15, 9, 10, 11, 42, 43>: fem+masc {fem masc} ;
NUM<15, 9, 10, 11, 42>: s-sg+pl {s-sg s-pl {s-pl-sp s-pl-unsp}} s-mass ;
PERS<15, 9, 10, 11>: non-3rd {1st 2nd} 3rd indet;
```

2.3 Morphology

In the morphology, we specify the words to be recognized by the grammar and the features related to them. For this reason, *all* entries defined in the morphology are provided with synctactical features, which serve to ensure e.g. the elements agreement or to select the argument forms allowed in the lexical definitions. However, dictionary entries might contain both semantical and synctactical features, if required:

```
word def-det: Det {
  il: masc s-sg sg unique specific def norm;
  lo: masc s-sg sg unique specific def cons;
  la: fem s-sg sg unique specific def norm;
  la: fem s-sg sg unique specific def cons;
  "l’": masc s-sg sg unique specific def voc;
  "l’": fem s-sg sg unique specific def voc;
  i: masc s-pl pl unique specific def norm;
```

```
  gli: masc s-pl pl unique specific def mod;
  le: fem s-pl pl unique specific def norm;
  le: fem s-pl pl unique specific def mod;
}
```

In the example above, the *number* value is defined twice in each entry: by the synctactical features (`s-sg` or `s-pl`), and the semantical ones (`sg` or `pl`). The first group allows us to control the agreement between the determiner and its arguments (e.g. nouns), which means: a determiner containing the feature `s-sg` will unify only with an argument containing the *same* feature or a *compatible* one (e.g. `s-sg+pl`), like in:

```
word gioco: Noun(abstract) {
  *: norm 3rd nom+acc masc s-sg norm;
  giochi: norm 3rd nom+acc fem s-pl norm;
}
```

The semantical features `sg` or `pl` are not relevant for the agreement control, but generate the additional node `<Num>` in the parse output. Similarly, the semantical features `unique` and `specific` will generate additional output nodes (see 2.4.2.2).

2.4 Lexicon

For the sake of clarity, we divided the lexicon section of the current Italian grammar in three main groups:

- Discourse units (DU): isolated utterances, discourse marker;
- Atomic categories: nouns, noun phrases etc.;
- Complex categories: complete verb-based utterances.

2.4.1 Discourse units

A very important task to make the interaction between robot and child as natural as possible, is to provide the system with a repertoire of general *set phrases* like greeting, feedback or even interjection-like expressions; the robot should be able to introduce itself, catch the child's interest and lead over the game phase.

Some *small talk* or *feedback* elements as well as colloquial *filler* can help to sustain the game phase (e.g. encouraging shy or wary children) and to bypass misunderstandings, technical problems or other difficulties. Therefore we collected in this section a broad range of typical expressions and fillers and implemented them as Discourse unit.

Discourse units do not come into consideration as argument for a complex category, but can become part of an utterance, if an appropriate rule for DU concatenation is activated. For more details about Discourse units concatenation, see [3].

The most of these expressions consist of a single word; for those consisting of more as one word, we had to define separated lexical categories to avoid overgeneration.

```
word ehi: Greet(greeting);

word dai: DU(cue);
```

```
word oh: DU(cue);
word okay: DU(cue);
word oops: DU(cue);
word accidenti: DU(cue);
word caspita: DU(cue);

word grazie: Thanks;

family DU {
  entry: du<15>[T]: T (*);
}

family Thanks(DU) {
  entry: s<10>[E 2nd s-sg] \ np<15>[T nom]: E (* <Actor>T);
  entry: s<10>[E 2nd s-sg] \ np<15>[T nom] / (s[V inf] \ np[T]) / prep<4>[S stem=per]:
       E (* <Actor>T <Subordinate>(S:m-final ^ <Anchor>V));
}
```

Also other word classes (Adjectives, Adverbs, even some Nouns, see below) can serve as reference to form such expressions. These must be defined each in a dedicated lexical category. To avoid conflicts, the *main part of speech* must be specified besides the *family declaration*; otherwise, the word will not work properly in other lexical categories.

```
family Feedback(Adv) {
  entry: du<25>[M]: M (*);
  member: ancora bene benissimo male;
}

family NotBad(indexRel="Modifier" Pron) {
  entry: du<15>[T] /^ adv<15>[T stem=male]: T (<Modifier> *);
  member: niente;
}

family Please(indexRel="Anchor" Noun) {
  entry: du<15>[T] \^ prep<~15>[T stem=per]: T:m-comment (<Anchor> *);
  member: favore piacer_noun;
}
```

2.4.1.1 Punctuation marks

As they give the TTF modules important guidelines for the utterance prosody and intonation, punctuation marks build a particular lexical category in this grammar. In some cases we used them to differentiate similar categories, so that their results can be integrated in an appropriate context.

```
family DM(indexRel="*NoSem*") {
  entry: dm<15>[T stem=*];
}

# ciao
family Greet(DU) {
  entry: s<10>[E unkn]: E (*);
  entry: s<10>[E 2nd s-sg GEN unkn] / adr<15>[X GEN stem=amico] /^ dm[stem=comma]:
       E (* <Addressee>X);
}
```

2.4.2 Atomic categories

The atomic categories are the *basic* components of the Lexicon (NP, PP etc.), which can be combined to form complete verb-based utterances. In addition to the *classical* categories (adjectives, names etc.), in this section we also defined more complex elements. Doing so, we can cover plenty of utterance variants with a single category and thus simplify the maintenance of the grammar.

Above these, we implemented a number of lexical categories to cover domain-specific issues. As mentioned above, some of them are declared as Discourse units.

In some *secundary* categories (e.g. determiners, particles), which are thought to be re-used in more complex entities, we decided to omit the category output completely. To ensure a correct realization, we introduced the value `indexRel="VALUE"` in the category index to mark the missing output.

The specified value corresponds to the relevant semantic feature specified for the category (e.g. `VOICE<E:Voice>: active passive reflexive;`). The value `indexRel="*NoSem*"` is used instead, when the defined category doesn't have a relevant semantic feature.

```
# mi
family ReflPron(indexRel="Voice" Pron) {
  entry: refl<10>[E OBJ];
}

# si
family ImpersPron(indexRel="*NoSem*" Pron) {
  entry: pn<15>[T];
}
```

Doing so, in the output of the main definition re-using these categories will not appear the category stem, but the semantic feature related with it – e.g. the *semantic role* of the reflexive pronoun (`<Voice>`), as in the utterance *mi chiamo Nao*.

```
family IntrodV(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom] \^ refl<10>[E dir] / np<3>[Y]:
       E (* <Actor>T <Patient>Y:person-name);
}

@chiamare1_0:action(chiamare ^
                    <Mood>ind ^
                    <Tense>pres ^
                    <Voice>reflexive ^
                    <Actor>(n1_0:entity ^ 1st ^
                            <Num>sg) ^
                    <Patient>(nao1_0:person-name ^ Nao) ^
                    <Subject>n1_0:entity)
```

2.4.2.1 Common and proper names

All words defined in the morphology are provided with syntactical features, which serve to ensure the elements agreement or to select the argument forms allowed in the lexical definitions. As nouns also unify with verbs when acting as subject, a feature for person was defined as well. Common and proper names usually take the 3rd person.

```
GEN<15, 10>: fem+masc {fem masc} unkn;
NUM<15, 10>: s-sg+pl {s-sg s-pl {s-pl-sp s-pl-unsp}} s-mass ;

word gioco: Noun NounSu{abstract} {
  *: 3rd s-sg nom+acc masc norm;
  giochi: 3rd s-pl nom+acc masc norm;
}
```

As mentioned above, we took not only simple nouns such as *gioco* into consideration, but also complex entities, i.e. *gioco delle categorie* or *gioco nuovo*. For this reason, we defined several categories covering the most common noun-based combinations, which can occur as subject or object in a verbal phrase. According with the *classical* grammar definitions, we marked the result category as N. This is relevant, since usually only noun phrases (NP) combine with verbs to form complex categories, while N combines e.g. with determiners to form NPs.

```
# gioco
family Noun {
  entry: n<15>[T]: T (*);
}

# gioco delle categorie
family NounOf(Noun) {
  entry: n<15>[T] /^ pp<3>[Y]: T (* <Compound>Y:m-specifier);
}
```

Common names and noun phrases usually don't generate an output in this grammar, unless they are bounded in complete verb-based sentences or *explicitly allowed as Discourse units*. Among others, we choose the latter solution to cover the answers and listing issues used in the PAL Quiz game.

```
# quiz: la risposta numero due
family ListN {
  entry: lst<15>[T fem s-sg norm] \ n<3>[Y s-sg stem=risposta] \^ n<4>[Z stem=numero]:
       T (* <Ref>Y <Ref2>Z);
  entry: lst<15>[T fem s-sg norm] \^ n<3>[Y s-sg stem=numero]: T (* <Ref>Y);
  entry: lst<15>[T fem s-sg norm] \^ n<3>[Y s-sg stem=risposta]: T (* <Ref>Y);
  entry: lst<15>[T fem s-sg norm]: T (*);
}
```

As we want to re-use *only* these category results – and not every DU – in other expressions of the Quiz game, we provided them with a specific shortcut.

```
# la risposta è ...
family AnswerV(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom stem=risposta] / answ<3>[Y nom]:
       E (* <Actor>T <Modifier>Y);
  entry: s<10>[E VFORM unkn] \ np<15>[T nom stem=risposta] / lst<3>[Y]:
       E (* <Actor>T <Modifier>Y);
  member: essere;
}
```

However, they can output as DU as well using a dedicated CCG rule.

```
rule {typechange: answ[D x2du] => du[D];}
```

2.4.2.2 Determiners

As determiners are *noun modifier*, we assigned to the result category the same semantic identifier as the N reference (`[T]`), so that its semantical features become subnodes of the noun output. Giving the result and its argument the same ID value, we ensured that both feature structures unify completely. The result category is a noun phrase (NP).

Note that the determiner stem is omitted in the output; its semantic role (e.g. *specificity*) is expressed instead by the semantical features defined in the morphology for this category.

```
# la risposta
family Det(indexRel="Delimitation") {
  entry: np<15>[T TYPE] /^ n<15>[T];
}

@risposta1_0:abstract(risposta ^
                      <Delimitation>unique ^
                      <Num>sg ^
                      <Quantification>specific)
```

For masculine nouns, two determiners are allowed in Italian, depending on the first letter of the very next word: singular *il* and *lo* and plural *i* and *gli*. In order to prevent wrong combinations, the additional feature AGREE was added. Similar to the features GEN and NUM, it ensures that only components with the same values can unify.

```
AGREE<15, 9>: norm mod {voc cons};

word diabete: Noun(abstract): 3rd nom+acc masc s-sg norm;

word def-det: Det {
  il: masc s-sg sg unique specific def norm;
  lo: masc s-sg sg unique specific def cons;
}
```

2.4.2.3 Prenominal and postnominal adjectives

Also adjectives are *noun modifier*; we assigned to the result category the same semantic identifier as the N reference, so that its semantical features becomes subnodes of the noun output. Giving the result and its argument the same ID value, we ensured that both feature structures unify completely. The result category is N, as it should combine with a determiner to form a NP.

In the majority of cases, attributive adjectives are postnominal in Italian. Some can change their position instead (e.g. *bello*, *buono*): although they might follow the noun in some contexts, they are mostly prenominal. In this case, they take specific inflection forms, which are not allowed in other contexts: for example *un bel gioco*, but *un gioco bello*.

The selection of the appropriate form can be easily achieved as we did for the determiners, i.e. introducing the AGREE value for prenominal adjectives.

```
word bel: Adj_pre_only(quality pred=bello) {
  *: masc s-sg s-degree-base norm;
  bello: masc s-sg s-degree-base cons;
}
```

However, this also affects the noun-determiner agreement. As mentioned above, the determiner form depends on the first letter of the very next word, so that in some cases we have to allow different determiners for the same noun: *il gioco* but *lo stesso gioco*.

In this case, if we used the standard definition (requiring the feature structures of noun and adjective to unify completely), we would allow only combinations with the same AGREE value – i.e. the wrong phrase **il stesso gioco*.

To avoid this, the *partial inheriting* was used for prenominal adjectives: `<~15>`. Using this construct, the result category inherits all the features from the argument `<15>` as usual – with exception of AGREE, which is overridden with an explicit feature value, so that the result category can combine only with a determiner having the same value.

```
# bel nome
family Adj_pre_only(indexRel="Modifier" Adj) {
  entry: n<~15>[norm] /^ n<15>[T]: T (<Modifier>(M *));
}
```

In some cases, nouns can be dropped in Italian – e.g. when giving an answer or talking about known objects. For a better context definition, the dropped nouns are then often substituted by their modifier, like in the expression *scelgo la prima*. If we want this *contextualized* modifier to be used as an argument of a verbal phrase, we have to treat it as a normal **noun phrase**.

In this case, we express the semantic role of the dropped noun with the pre-defined head *context* [3]. Doing so, the modifier can be displayed in the correct feature `<Modifier>`.

```
# risposta giusta
family Adj {
  entry: noun<15>[T stem=context norm 3rd]: T:entity (context <Modifier>(M *));
}

@context1_0:entity(context ^
                   <Delimitation>unique ^
                   <Gen>fem ^
                   <Num>sg ^
                   <Quantification>specific ^
                   <Modifier>(primo1_0:number-ordinal ^ primo))
```

Above this, we implemented a broad range of adjectives as Discourse units to cover domain-specific issues – among others, *feedback* expressions.

```
# ottimo lavoro
family Feedback3(indexRel="Modifier" Adj) {
  entry: du<~15>[voc] /^ n<15>[T s-sg]: T (<Modifier>(M *));
  member: ottimo;
}

# bravo
family Feedback4(Adj) {
  entry: s<10>[E 2nd s-sg] \ np<15>[T nom]: E (* <Actor>T);
}
```

2.4.2.4 Adverbs

Often adjective comparatives or superlatives are built using special adverbial modifier in Italian, which are not allowed in other contexts. We thus provided the result category with a special shortcut `pred`, in order to avoid overgeneration and wrong combinations. By specifying the feature `base` in the argument, we allowed only *positive* adjectives to combine with the modifier.

```
# molto
family SupAdv(indexRel="Degree" Adv) {
  entry: pred<~25>[M] /^ pred<25>[M s-degree-base];
}
```

Other simple adverbs or adverbial expressions were defined as *verbal phrase* modifiers (`s`).

```
# di nuovo
family Iter3(indexRel="Modifier" Prep) {
  entry: s<10>[E] \ np<15>[T nom] \* (s<10>[E fin] \ np<15>[T])
       / pred<25>[A stem=nuovo masc s-sg s-degree-base]:
       E (<Modifier>(I:m-time-sequence * <Anchor>A));
  member: di;
}
```

2.4.2.5 Prepositional phrases

Prepositions usually modify NPs to form a prepositional phrase (PP). According to the MOLOKO grammar, prepositions build the *semantic head* of these categories, while the nouns they introduce are represented in the node `<Anchor>`. Consequently, the category output takes the semantics of the preposition, as it can't inherit the semantics of the noun anymore.

However, we need exactly these semantic definitions to *select* the PP dependencies in complex categories. To solve this, we implemented several *specific* lexicon categories, instead of a generic PP category. For each preposition, we defined a new family containing more *categories*, where the semantic output of each entry depends on the semantics of the noun introduced by the preposition.

```
family PrepA(Prep) {
  entry: pp<~15>[T] /^ np<15>[A GEN acc]: T:m-benefactor (* <Anchor>A:animate);
  entry: pp<~15>[T] /^ np<15>[A GEN acc]: T:m-result (* <Anchor>A:abstract);
  entry: pp<~15>[T] /^ np<15>[A GEN acc]: T:m-result (* <Anchor>A:thing);
  member: a;
}
```

Of course, we can re-use the same semantics for phrases introduced by other prepositions, as long as they express the same concept. Once the wished PP combinations are defined properly, we can easily select the PP dependencies of a new category using their semantics.

Note that a very typical phenomenon in Italian is the *fusion* of a preposition with the following definite determiner: *il gioco di+le domande = il gioco delle domande*.

This concerns the most prepositions, but *not all* of them. Above this, some NPs such as proper names don't require determiners, so they should combine only with simple prepositions – as well as some location descriptions or NPs with indefinite determiner.

The corresponding lexical categories must thus take all these variants into account. Again: the only way to manage this, is to define them in different families. Semantics might help to achieve a fine-grained context differentiation.

```
# delle domande
family PrepDi+det(Prep) {
  entry: pp<~15>[T stem=*] /^ n<15>[A GEN acc]: T:m-condition (* <Anchor>A);
}
```

2.4.2.6 Pronouns

We distinguish in this grammar five main pronoun types: personal, reflexive, possessive, demonstrative and relative. The most of them were defined in a similar way; however, in order to avoid overgeneration or wrong combinations, we provided them with a TYPE feature, which can be used in the complex categories to select the pronoun type needed.

```
TYPE<15, 9>: def indef demo perso card;
```

While the result category of demonstrative and personal pronouns is NP, we assigned N to the possessive pronouns, because in Italian they should combine with a determiner to form a NP.

```
family Pron {
  entry: np<15>[T]: T (*);
}

family PossPron(indexRel="Owner" Pron) {
  entry: n<~15>[norm] /^ n<15>[M]: M (<Owner> *);
}
```

As they serve as verbal phrase modifier, the relative pronouns require a subject-dependent (resp. object-dependent) VP argument. To ensure the verb-subject agreement, we provided the corresponding items in the VP argument with the ID values supported by the type definition of PERS and NUM. The result category is N.

```
PERS<15, 10, 11>: non-3rd {1st 2nd} 3rd indet;
NUM<15, 10, 11>: s-sg+pl {s-sg s-pl {s-pl-sp s-pl-unsp}} s-mass ;

# movimenti che conosciamo
family RelPron(indexRel="Role-in" Pron) {
  entry: n<~15> \* n<15>[T] / (s[E s-ind] \ np[X] / np[T]): T (<Role-in>E);
}
```

2.4.2.7 Conjunctions

Usually we define conjunctions as isolated discourse unit (e.g. *chissà*, *magari*, *purtroppo*, see 2.4.1) to enhance the *colloquial* character of the verbal output and make it more natural. Doing so, we also simplify the recognition of even very complex sentences, as the conjunctions can bind fully independent utterances.

However, in the case of the coordinating conjunction *e*, we preferred a complex category, to avoid the concatenation of NPs having different cases (especially pronouns).

```
# le braccia e le gambe
family Coord(indexRel="*NoSem*" ConjDU) {
  entry: np<~15>[T s-pl] \* np<15>[X1 nom] /* np[X2 nom]: T (* <First>X1 <Next>X2);
  entry: np<~15>[T s-pl] \* np<15>[X1 acc] /* np[X2 acc]: T (* <First>X1 <Next>X2);
}
```

2.4.2.8 Other atomic categories

Several other categories were introduce to adapt the grammar coverage to the canned-text rules output. Among others:

- We implemented a number of indefinite pronouns, e.g. requiring only a singular resp. plural N argument (*ogni*, *tutti*, *molti*) and enhanced the coverage precision of demonstrative and interrogative pronouns. The indefinite pronouns *chiunque*, *qualcosa*, *qualcuno* can now serve as subject NP as well as the expression *anch'io*.
- The adverb *senza* also requires a plural N argument; the category result was marked as PP to ease its re-use in the complex lexical categories.
- We added adverbs which are often part of feedback or colloquial expressions (e.g. *proprio*, *decisamente*, *pochino*) and extended the temporal adverb section (*oggi*, *domani*, *stavolta*).
- We implemented particles to cover reflexive verbs (*mi chiamo ...*), as well as partitive (*ne ho indovinate tre*) and pronominal (*ce l'ho*) particles.

2.4.3 Complex categories

This is the *main* grammar section. Here we can find the definitions for complete verb-based utterances. As a general rule, verbs serve as reference word for these categories, while the atomic categories described above are used as arguments, to form syntactically correct structures.

2.4.3.1 Transitive and intransitive verbs

To ensure the subject-verb agreement of these categories, verb and subject NP are both associated with the type definition of PERS and NUM over their ID. Thus we usually don't have to specify their features explicitly.

```
NUM<15, 10>: s-sg+pl {s-sg s-pl {s-pl-sp s-pl-unsp}} s-mass ;
PERS<15, 10>: non-3rd {1st 2nd} 3rd indet;

# giochiamo
family IntransV(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom]: E (* <Actor>T);
}
```

The most verb-based utterances require normally NPs or PPs as argument. We used semantics to select the word sets allowed for each single argument, where required.

```
family DitransV(V) {
  entry: s<10>[E uninfl unkn] \ np<15>[T nom] \^ np<3>[Z dat dir] / np<4>[Y acc]:
       E (* <Actor>T <Patient>Y <Recipient>Z:animate);
```

Above this, we implemented some verbal clauses variants like *comincio io* (subject has a different position) and *vuoi giocare di nuovo?* (action iteration). The first variant is the only case in which the typical Italian pro-drop doesn't make sense, as the pronoun is *emphasized* over its position at the end of the sentence and should not be dropped in any case. As the second variant is already covered by the standard pro-drop rules, we don't need further ones.

```
# comincio io
family IntransInvV(V) {
  entry: s<10>[E VFORM unkn] /^ np<15>[T perso nom]: E (* <Actor>T);
}
```

2.4.3.2 Compound verbs

Compounds with an auxiliary verb (such as present or past participle and modal verbs) require a verbal phrase (VP) as argument. The standard shortcut for this, `(s\np)`, means that *every* verbal phrase defined in the Lexicon can be combined with the reference verb – regardless of the real number of its arguments. To avoid overgeneration, we have therefore to specify at least the verb form in the VP argument:

```
VFORM<10>: fin inf prpt ppt;

# vuoi fare un' altra domanda
family ModV(indexRel="Aux" V) {
  entry: s<10>[E uninfl unkn] \ np<15>[T nom] /^ (s[E inf] \ np[T]): E (<Aux>(V *));
}
```

Of course, we could also define the VP argument as *atomic* category – similar to NPs. However, complex categories should be preferred in this case, because:

- Main and auxiliary verbs should have the same actor. Therefore, both subject NPs receive the same semantic identifier `[T]` and morphological values, if required.
- In the output structure, the actor should be assigned to the `<Event>` node – and not to the auxiliary verb node.

From a syntactical point of viev, the *main* verb should be expressed by the stem of the VP argument, as the compound verbs are usually just different verbal tenses. For this reason, we defined the auxiliary verbs as a *subnode* of the VP argument (`<Aux>`). All defined dependencies belong then consequently to the main verb.

```
@fare1_0:action(fare ^
               <Mood>ind ^
               <Tense>pres ^
               <Actor>(n1_0:entity ^ 2nd ^
                       <Num>sg) ^
               <Aux>(volere1_0:action ^ volere) ^
               <Patient>(domanda1_0:abstract ^ domanda ^
                         <Delimitation>existential ^
                         <Num>sg ^
                         <Quantification>specific ^
                         <Modifier>(altro1_0:quality ^ altro)) ^
               <Subject>n1_0:entity)
```

2.4.3.3 Inflected past participles

A special case is the past participle of **essere**. While non-finite verb forms are usually uninflected, GEN and NUM of these participles unify with the subject NP, e.g. *sei stato bravo*.

However, verbs (and thus VP arguments) have only PERS and NUM – not a GEN value, so we had to specify it explicitly in this case. To make it unify with the subject NP gender, we defined the same feature there. This works only if the VP argument has one of the ID values supported by the type definition of GEN:

```
GEN<15,10,42>: fem+masc {fem masc} unkn;

family CopV(indexRel="Aux" V) {
  entry: s<10>[E VFORM GEN] \ np<15>[T nom GEN] /* (s<42>[E GEN ppt reg] \ np[T nom]):
       E:spec (<Aux>(V *));
}
```

The past participle of **avere** is usually uninflected, except in case the direct object is a pronoun: *ho indovinato la risposta* but *l'ho indovinata*. Unlike *essere*, these participles unify with the object. This required separate entries for inflected and uninflected participles, which differ in the explicite feature `infl` resp. `uninfl`. Of course, the same features should also be defined in the result categories of the verbs the VP arguments is referred to.

```
family AuxV(indexRel="Aux" V) {
  entry: s<10>[E s-ind unkn] \ np<15>[T nom] /^ (s[E masc s-sg ppt uninfl] \ np[T]):
       E:action (<Aux>(V *));
  entry: s<10>[E s-ind unkn] \ np<15>[T nom] \^ pron<3>[Y acc GEN NUM] /^
       (s[E GEN NUM ppt infl] \ np[T] \ pron[Y]): E:action (<Aux>(V *));
}
```

The pronoun should **not** unify with the subject and verb inflection in this case. For this reason, we had to choose for `pron` an ID value *not supported* by the type definition of GEN or NUM. This makes it independent of the verb-subject agreement, so that we can freely define there the features ensuring the unification with the VP argument.

2.4.3.4 Final verbs

We defined this category using its keywords (*per*, *a*) as reference. Doing so, we could cover with a single category all possible VP combinations, e.g. *servono per crescere*. Otherwise, we should provide *each* utterance definition with an additional entry for the corresponding final phrases.

```
family FinalV(indexRel="Subordinate" Prep) {
  entry: s<~10> \ np[T nom] \^ (s<10>[A fin] \ np<15>[T]) /^ (s[V inf] \ np[T]):
       A (<Subordinate>(E:m-purpose * <Anchor>V));
}
```

However, we had to use the *inheriting* in the result category (`<~10>`), because the reference words don't have these values themselves. To make the result inherit these values from the VP argument, we defined the same ID there (`<10>`).

In order to ensure the unification with the finite VP argument, we assigned to the subject NP and the corresponding items in the VP argument the ID values supported by the type definition of PERS and NUM. The result category is a verbal phrase (S).

2.4.3.5 Verbs with prepositional or subordinate phrases

Some verbs are allowed only with a PP (*pensare a* + noun), a prepositional VP (*sperare di* + infin) or a subordinate clause (*dire che* + fin) as argument. We could not define them using the introducing keywords as reference, because these verbs have no predefined category to which we can refer. So we defined them like normal verbs, and specified the introducing words separately.

```
# passiamo al diabete
family PrepVA(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom] / pp<3>[Z]:
       E (* <Actor>T <Patient>Z:m-result);
}
```

The prepositional VP argument was defined using the complex shortcut `(s\np)` as for compound verbs (see above). In order to avoid overgeneration, we selected its verb form explicitly.

```
# spero di rivederti
family InfDi(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom] / (s[V inf] \ np[T]) / prep<4>[S stem=di]:
       E (* <Actor>T <Subordinate>(S:m-comment ^ <Anchor>V));
}
```

Verbs introducing a subordinate clause were defined similarly. However, in this case we preferred an *atomic* VP argument, so that also subordinate clauses with a different subject can be recognized: e.g. *penso che sia la numero tre*.

```
family SubV(V) {
  entry: s<10>[E VFORM unkn] \ np<15>[T nom] / s[V fin] / sb<4>[V]:
       E (* <Actor>T <Subordinate>V);
}
```

2.4.3.6 Negative and interrogative clauses

We defined these categories using their typical keywords as reference; they require a complex VP argument. Also in this case, the negation or interrogative particle is expressed as subnode of the main VP clause; the negation introduces the node `<Polarity>` over its semantical features.

```
# non ho indovinato
family Negation(indexRel="Polarity") {
  entry: sn<10>[E] \ np<15>[T nom] /* (s<10>[E fin] \ np<15>[T]);
}

# chi fa le domande
family IntSubj(indexRel="Wh-Restr" Pron) {
  entry: si<10>[E] / (s<10>[E s-major] \ np[T]): E (<Wh-Restr>(T:animate *));
}
```

2.4.3.7 Infinitive and Imperative with embedded pronouns

In Italian, direct and indirect pronouns following an infinitive or imperative verb form are attached to the end of it: *lo ripeto* vs. *voglio ripeterlo* and *ripetilo*. While imperative forms can work alone, infinitive ones can be used only as argument of a verbal phrase.

For the sake of precision, we defined different categories to delimit the exact context in which these forms can be used: while forms containing direct pronouns don't need other arguments (e.g. *ripetilo*, *sono contento di rivederti*), forms with indirect pronouns always require a further direct object (*dammi la mano*, *sai dirmi perché*); some forms expect a VP argument (e.g. *vuoi ripeterlo*); some others are reflexive (e.g. *non preoccuparti*) and therefore require different arguments then standard verb forms.

```
# imitarla
family InfDirObj(V) {
  entry: infd<10>[E inf]: E (*);
  entry: infd<10>[E inf] / pp<3>[Z]: E (* <Recipient>Z:m-benefactor);
}

# farti una domanda
family InfIndirObj(V) {
  entry: infi<10>[E inf] / np<3>[Y acc]: E (* <Patient>Y);
}

# preoccuparti
family ReflInfV(V) {
  entry: rinfi<10>[E inf stem=*]: E (*);
}
```

As noticed, we assigned to each category *type* a particular shortcut instead of the standard `s`. Every shortcut leads to a dedicated CCG rule block, where we can extract the embedded pronouns to the appropriate output features for each category (see 2.5.2 and 2.5.3).

2.5 CCG Rules

This section specifies the rules available for the current grammar. OpenCCG is provided with a standard set of rules: application, typeraising, and composition (harmonic as well as crossed) [4]. They control the interaction of the categories defined in the grammar; for example, without them we could not combine atomic categories into complex structures.

Usually, all standard rules are activated in OpenCCG, but we can use this section to switch off single sets of rules for the current grammar. Further rules (e.g. typechange) can be added, if required. Among other things, we can use them to simplify the grammar maintenance – e.g. automating constantly recurring structures.

2.5.1 Pro-drop rule

A very interesting rule in this CCG grammar regulates automatically the typical Italian pro-drop: *sono Nao* instead of *io sono Nao*. Without this rule, we should define two variants for *each* verb entry: one with subject, one without it. Note that the rules developed for this purpose must use the feature IDs and semantic identifiers defined in the type hierarchy; otherwise, they would not work properly.

```
rule {
  typechange: s<10>[fin] \^ np<15>[nom] => s<10>[fin];
}
```

In the meanwhile we extended this rule to that effect, that the morphological features of the *dropped* subject NP are transmitted into the output structure. Further rules were developed to enrich the pronoun output nodes similarly. This enhanced the realization precision significantly; we might also use this information in future, to coordinate the agreement among concatenated discourse units (see 2.5.4).

```
rule {
  typechange: s<10>[E 1st unkn s-sg fin] \ np<15>[T nom] =>
  s<10>[E]: E (<Subject>(T:entity 1st <Num>sg));
}
```

As the subject is missing, the rule takes the relevant features from the verb and transmits them to a new output node, `<Subject>`. This node should share the semantic identifier `[T]` with the subject NP, so that the morphological features will nonetheless populate the subject output node `<Actor>`.

After this transformation, the output for the utterance *sono Nao* looks as follows:

```
@essere1_0:state(essere ^
                 <Mood>ind ^
                 <Tense>pres ^
                 <Actor>(n1_0:entity ^ 1st ^
                         <Gen>masc ^
                         <Num>sg) ^
                 <Patient>(nao1_0:person-name ^ Nao) ^
                 <Subject>n1_0:entity)
```

2.5.2 Direct and indirect object of imperatives

Similar rule blocks were developed to extract *embedded* pronouns from the imperative and infinitive compound verb forms.

The **imperative** verb forms (e.g. *mostrami un movimento*) are finite; implementing dedicated pro-drop rules, we can extract the subject feature values as we do for normal clauses.

```
rule {
  typechange: sind<10>[E o-1st o-sg] \ np<15>[T nom] =>
  s<10>[E] \ np<15>[T nom]: E (<Recipient>(Y:person 1st <Num>sg));
}

@mostrare1_0:action(mostrare ^
                    <Mood>imp ^
                    <Actor>(n1_0:entity ^ 2nd ^
                            <Num>sg) ^
                    <Patient>(movimento1_0:abstract ^ movimento ^
                              <Delimitation>existential ^
                              <Num>sg ^
                              <Quantification>specific) ^
                    <Recipient>(n22_0:person ^ 1st ^
                                <Num>sg) ^
                    <Subject>n1_0:entity)
```

As the rules results are marked as `s`, they can match in every lexical category requiring an imperative clause as argument, but also work alone like *full* finite sentences.

2.5.3 Direct and indirect object of infinitives

The **infinitive** verb forms (e.g. *rivederti*) obviously don't include inflection feature values, so that pro-drop rules are useless in this case. However, the lexical categories requiring a VP argument always expect them to include a subject NP.

```
# sono felice di rivederti
family PredV(V) {
  entry: s<10>[E VFORM GEN] \ np<15>[T nom GEN] / (s[V inf] \ np[T])
       / prep[S stem=di] /* pred<25>[M GEN]:
       E (* <Actor>T <Modifier>M <Subordinate>(S:m-comment ^ <Anchor>V));
}
```

As their standard lexical category doesn't include a subject NP, these verb forms wouldn't match in normal case. For this reason, we modified the dedicated rules to that effect, that the results contain a *fake* subject NP, which will inherit the semantic features of the *main clause* subject in the category output.

```
family InfDirObj(V) {
  entry: infd<10>[E inf]: E (*);
}

rule {
  typechange: infd<10>[E o-2nd o-sg] =>
  s<10>[E] \ np<15>[T]: E (<Actor>T:entity <Patient>(Y:obj-pron 2nd <Num>sg));
}

@essere1_0:state(essere ^
                 <Mood>ind ^
                 <Tense>pres ^
                 <Actor>(n1_0:entity ^ 1st ^
                         <Gen>fem ^
                         <Num>sg) ^
                 <Modifier>(felice1_0:q-attitude ^ felice) ^
                 <Subject>n1_0:entity ^
                 <Subordinate>(di1_0:m-comment ^ di ^
                               <Anchor>(rivedere1_0:action ^ rivedere ^
                                        <Actor>n1_0:entity ^
                                        <Patient>(n22_0:obj-pron ^ 2nd ^
                                                  <Num>sg))))
```

As noticed, we assigned to these rule outputs the standard shortcut for verbal clauses `s` as well. Doing so, we automatically allow these verb forms to match in all lexical categories requiring an infinitive clause as argument. However, they will not work alone, as the rules converting the *relevant* category results in Discourse units support only *finite* verb forms (see 2.5.4).

The verb forms with indirect pronouns were covered accordingly.

2.5.4 Discourse unit rules

Every category result defined in this grammar must be converted in a Discourse unit (DU) to generate an output. This gives us the possibility to recognize complex utterances, even if they are not *physically* defined in the grammar – as long as their constituents are specified there as lexical category.

Of course, not every category result should be allowed for concatenation, as there is no way to control the agreement among concatenated DUs, so that overgeneration and unwanted results cannot be excluded. For this reason, we disabled the rules concerning *non-finite* verb phrases or the most atomic categories, so that only *finite*, *complete verb-based sentences* and *specific DUs* generate an output and are allowed for concatenation.

```
rule {
  typechange: s[D x2du] => du[D];
  typechange: sn[D x2du] => du[D];
  typechange: si[D x2du] => du[D];
  typechange: answ[D x2du] => du[D];
}
```

After converting every relevant result category into DU (see 2.5.4), we can line up the DUs found in the grammar. The output is structured in a list.

```
rule {
  typechange: du[A] => du[B]/^du[C]: B:d-units (list <First>(A) <Next>(C));
}
```

2.6 Other matters

The implementation of the OpenCCG logical forms in the Content planner highlighted some points for improvement in the Italian OpenCCG grammar. The grammar has been completely overhault and reorganized, so that the most cases of unprecise or overgenerating parses could be solved.

For example, the semantics of word categories like Adjectives or Prepositions has been revised; some relevant categories were simplified, while we removed *unused* ones, which were developed in the first project phase. As the canned text was improved continuously, we adapted the coverage to the current version.

A similar effort was done for the English OpenCCG grammar, so that we don't expect many problems during the implementation of the logical forms in the Content planner for this language in the future.

3 LF grammar

The new LF grammar supports all the Speech acts and variants defined in the current canned text version. For illustration purposes, in this report we will focus on some representative examples.

3.1 Gender-dependent inflection

The gender-dependent inflection of adjectives and some verb forms (e.g. past participle) in Italian is a big issue in the canned text: In the original draft, the robot should have had the same gender as the child it is talking to, but in the long run the default value was set to *male*, to simplify the rule maintenance.

As for the child gender, we still have to define two different rules (or rule alternatives) for each relevant Speech act – one for male, one for female children.

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>answer)
->
# ^ <SpeechModus>interrogative

{
^ <Context>(<ChildGender>(unknown|m))
->
# ^ :canned ^ <stringOutput>random("okay ", "sei pronto ").

^ <Context>(<ChildGender>f)
->
# ^ :canned ^ <stringOutput>random("okay ", "sei pronta ").
}
```

In this context, it becomes clear how the use of *logical forms* and *features* instead of *strings* can make the parameterization of such variables more comfortable. As already said, the OpenCCG semantic output represents the parsed word(s) in the canonical form, and expresses their inflection generating appropriate features:

```
tccg> sei pronta
1 parse found.

Parse: du :
  @essere1_0:state(essere ^
                   <Mood>ind ^
                   <Tense>pres ^
                   <Actor>(n1_0:entity ^ 2nd ^
                           <Gen>fem ^
                           <Num>sg) ^
                   <Modifier>(pronto1_0:q-attitude ^ pronto) ^
                   <Subject>n1_0:entity)
```

Setting context-dependent variables for the relevant features of this logical form, in the LF grammar we could cover the gender-specific output with one simple rule.

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>answer)
^ <Context>(#ctxt:context ^ <ChildGender>#gen)
->
###a1 = :cue ^ okay ^ <Polarity>pos,
###a2 = :state ^ essere ^ <Mode>interrogative ^ <Mood>ind ^ <Tense>pres ^
        <Actor>(:entity ^ <Gen>#gen ^ <Num>sg ^ <Pers>2nd) ^
        <Modifier>(:q-attitude ^ pronto),

# ^ <Action>(random(###a1,###a2)).
```

The system includes the default value *masc*. During the interaction, the appropriate output can be generated changing this feature, if needed.

`request(answer ^ <ChildGender>masc)` → *sei pronto?*

`request(answer ^ <ChildGender>fem)` → *sei pronta?*

The robot gender was parameterized in the same way and is currently set to *masc*, but this can be easily changed, or switched to match the child gender in the future.

3.2 Verbal output for Group interaction

As some experiments for group interaction were planned, we completed the relevant Speech acts with further alternatives, implementing the strings for singular or plural inflection. Also in this case, we had to differentiate the variants by child gender to get the correct inflection.

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>answer)
->
# ^ <SpeechModus>interrogative

{
^ <Context>(<ChildGender>(unknown|m) ^ <PlayerNumber>sg)
->
# ^ :canned ^ <stringOutput>random("okay ", "sei pronto ").

^ <Context>(<ChildGender>f ^ <PlayerNumber>sg)
->
# ^ :canned ^ <stringOutput>random("okay ", "sei pronta ").

^ <Context>(<ChildGender>(unknown|m) ^ <PlayerNumber>pl)
->
# ^ :canned ^ <stringOutput>random("okay ", "siete pronti ").

^ <Context>(<ChildGender>f ^ <PlayerNumber>pl)
->
# ^ :canned ^ <stringOutput>random("okay ", "siete pronte ").
}
```

Again, setting context-dependent variables for the relevant features of this logical form, we could cover all variants with one simple rule in the LF grammar. The default value *sg* was set, but this can be easily changed to *pl* in case of a group interaction.

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>answer)
^ <Context>(#ctxt:context ^ <ChildGender>#gen ^ <PlayerNumber>#n)
->
###a1 = :cue ^ okay ^ <Polarity>pos,
###a2 = :state ^ essere ^ <Mode>interrogative ^ <Mood>ind ^ <Tense>pres ^
        <Actor>(:entity ^ <Gen>#gen ^ <Num>#n ^ <Pers>2nd) ^
        <Modifier>(:q-attitude ^ pronto),

# ^ <Action>(random(###a1,###a2)).
```

3.3 Context-related contents

Some brilliant solutions implemented in the canned text were kept and improved in the LF grammar, like the use of context-related contents (e.g. game-specific verbs or elements).

The Content planner supports defined Speech acts for two different games: quiz, and sorting. Some utterances can be used in all games, but of course they must fit to the given context (e.g. familiarity: *ti va di giocare ancora?* vs. *ti va di giocare?*), so we should define different rules or rule variants for each one of these contexts. To simplify this, we introduced in the canned text particular features containing context-dependent values.

Also the LF grammar includes these context-related contents. Similarly, we also implemented further contents like game nouns (*gioco delle domande* vs. *categorie*) or – for other contexts – some *quantity expressions* (see 3.4).

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>play ^ !<GameNoun>)
{
^ <Context>(<CurrentGame>quiz)
->
#ctnt ^ <GameNoun>domanda.

^ <Context>(<CurrentGame>sorting)
->
#ctnt ^ <GameNoun>categoria.

^ <Context>(<CurrentGame>unknown)
->
#ctnt ^ <GameNoun>gioco.
}
```

In both cases, these contents can be set as a simple *variable* in the main rule. This allows us to cover more context-dependent utterances with just one definition.

```
:dvp ^ <SpeechAct>request
^ <Content>(#ctnt:content ^ <About>play ^ <InfVerb>#v)
^ <Context>(#ctxt:context ^ <RobotGender>#gen)
^ !<Action>
->
###a1 = :action ^ vedere ^ <Polarity>neg ^
        <Patient>(:e-time-unit ^ ora ^ <Delimitation>unique ^ <Num>sg ^
                  <Quantification>specific),
###a2 = :state ^ essere ^ <Modifier>(:q-attitude ^ impaziente),
```

```
# ^ <Action>(random(###a1,###a2): ^ <Mood>ind ^ <Tense>pres ^
             <Actor>(###act1:entity ^ 1st ^ <Gen>#gen ^ <Num>sg) ^ <Subject>###act1: ^
             <Subordinate>(:m-comment ^ di ^
                           <Anchor>(:action ^ #v ^ <Actor>###act1: ^
                                    <Patient>(:m-accompaniment ^ con ^
                                              <Anchor>(:person ^ 2nd ^ <Num>sg))))).
```

Again: the OpenCCG surface realizer will pick up the *variable* (e.g. the verb or game name) belonging to the defined constraint and generate the appropriate output.

3.4 Re-usable "chunks"

Another aspect borrowed from the canned text grammar is the implementation of re-usable *sentence chunks*. Under these we understand frequent set phrases (e.g. *per favore*, *mi dispiace*, *scusa*), which can also be Speech act-dependent, as well as context-dependent sentence parts, often subordinates, e.g. arguments of a generic main sentence:

Non vedo l'ora di ...

`<CurrentGame>quiz ^ <Familiarity>yes` → *... giocare ancora.*

`<CurrentGame>sorting ^ <Familiarity>no` → *... giocare con te.*

Also some *quantity expressions* belong to this category, like in the Speech act **provide(performance)**.

In questo giro hai dato ...

`<Performance>3` → *... tre risposte esatte.*

`<Performance>3 ^ <TotalCount>5` → *... tre risposte esatte su cinque.*

In this case, we had to take several variants into account: e.g. the performed *action* takes different genders in Italian, influencing the adjective inflection, while other variants are determined by the number of the performed actions (singular vs. plural, positive vs. negative), the asker role (robot vs. child), and the performance context (overall vs. game).

In the canned text, this required several rules or rule variants, as some of the relevant differences (e.g. *tre risposte esatte su cinque*) can't be just appended at the end of an existing utterance, but must take account of the synctactical correct word order.

Defining appropriate sentence *chunks* in the LF grammar, it was possible to reduce significantly the rule redundancy for this Speech act. In fact, LFs can be added in any position to every defined feature or subfeature; the correct word order will be determined by the OpenCCG rule which controls the appropriate semantic output.

So we defined *chunk* sets to differentiate the asker role and the performed actions, while a third chunk set generates an additional LF in case a `<TotalCount>` variable is set (*tre risposte su cinque*). Otherwise, these subfeatures will just not appear in the output.

Please note that in the chunk set for *performance* we use the game-dependent feature `<GameNoun>` (see 3.3).

```
// asker
:dvp ^ <SpeechAct>provide
^ <Content>(#ctnt:content ^ <About>(performance|performanceSession))
```

```
^ !<Actr>
{
^ <Context>(#ctxt:context ^ <Asker>user)
->
# ^ <Actr>(:entity ^ 1st ^ <Num>sg).

^ <Context>(#ctxt:context ^ <Asker>robot)
->
# ^ <Actr>(:entity ^ 2nd ^ <Num>sg).
}

// performance
:dvp ^ <SpeechAct>provide
^ <Content>(#ctnt:content ^ <About>(performance|performanceSession) ^ <GameNoun>#g)
^ !<Object>
{
^ <Content>(<Performance>0)
->
# ^ <Object>(:abstract ^ #g ^ <Delimitation>existential ^ <Num>sg).

^ <Content>(<TotalCount>#c: ^ <Performance>1)
->
# ^ <Object>(:abstract ^ #g ^ <Delimitation>existential ^ <Num>sg ^
             <Quantification>specific).

^ <Content>(<QuantityLF>#q: ^ <Performance>(!1^!0))
->
# ^ <Object>(:abstract ^ #g ^ <Delimitation>existential ^ <Num>pl ^
             <Quantification>specific ^ <Quantity>#q:).
}

// ... su ##
:dvp ^ <SpeechAct>provide
^ <Content>(#ctnt:content ^ <About>(performance|performanceSession) ^ <Overall>#t:)
^ <Object>(#obj:)
->
#obj ^ <Subgroup>(:m-comment ^ su ^
                  <Anchor>(:abstract ^ context ^ <Delimitation>existential ^
                           <Num>pl ^ <Quantification>specific ^ <Quantity>#t:)).
```

Finally, the *main* rule just collects the defined chunks and *distributes* them to the relevant features.

```
:dvp ^ <SpeechAct>provide
^ <Content>(#ctnt:content ^ <About>(performance|performanceSession) ^ <InfVerb>#v
            ^ <TotalRef>#tr:)
^ <Context>(#ctxt:context ^ <Asker>(user|robot))
^ <Actr>(#act:) ^ <Object>(#obj:)
^ !<Action>

{
// in questo giro ho/hai dato/fatto ...
^ <Content>(<Performance>(!0))
->
# ^ <Action>(:d-units ^ list ^
             <First>#tr: ^
```

```
                <Next>(:action ^ #v ^ <Aspect>perfect ^ <Mood>ind ^ <Tense>pres ^
                       <Actor>#act: ^ <Aux>(:action ^ avere) ^ <Subject>#act: ^
                       <Mode>affirmative ^
                       <Patient>(#obj: ^ <Modifier>(:quality ^ random(esatto, giusto))))).

// in questo giro non ho/hai dato/fatto
^ <Content>(<Performance>0)
->
# ^ <Action>(:d-units ^ list ^
             <First>#tr: ^
             <Next>(:action ^ #v ^ <Aspect>perfect ^ <Mood>ind ^ <Tense>pres ^
                    <Actor>#act: ^ <Aux>(:action ^ avere) ^ <Subject>#act: ^
                    <Mode>affirmative ^ <Polarity>neg ^
                    <Patient>(#obj: ^ <Modifier>(:quality ^ random(esatto, giusto))))).
}
```

The OpenCCG realizer will then output the surface corresponding to the chosen parameters, following the synctactical order defined in the OpenCCG Grammar.

3.5 Combining LF and strings

In some cases, it doesn't make any sense to convert a string to LF: the questions and answers included in the quiz game database were implemented in the LF grammar using the discourse unit concatenation. Doing so, LF and string can be easily combined. The following example was simplified for illustration purposes.

```
// ecco la ... domanda: [question]?
:dvp ^ <SpeechAct>provide
^ <Content>(<About>(question | questionRepeat) ^ <QuestionREX>#qrex:)
^ <Context>(<Question>#q)
^ !<Action>
->
# ^ <Action>(:d-units ^ list ^
             <First>(:cue ^ ecco ^ <Patient>#qrex: ^ <Mode>explicative) ^
             <Next>(:canned ^ <string>#q ^ <Mode>interrogative)).
```

3.6 Punctuation

Punctuation marks give the TTS system important guidelines for the utterance prosody and intonation. In the LF grammar, punctuation marks are generated automatically by defining for every output LF the feature `<Mode>` with an appropriate value (*interrogative* and *affirmative*, but also *explicative* for colon or *coordinative* for comma) – e.g. *arrivederci!*

```
:dvp ^ <SpeechAct>closing
^ <Context>(#ctxt:context ^ <CurrentGame>unknown)
^ !<Action>
->
###c1 = :greeting ^ ciao,
###c2 = :closing ^ arrivederci,

# ^ <Action>(random(###c1,###c2): ^ <Mode>exclamative).
```

The only exception is given by the Speech acts **accept**, **provide**, and **confirm** (e.g. role). As in Italian interrogative and affirmative utterances have the same word order, we can define just one rule for these Speech acts. However, the LF output doesn't include any mode feature, so that the correct punctuation mark is generated by corresponding Speech act-specific rules – e.g. accept: *va bene, domando io.* vs. confirm: *ho capito bene? domando io?*

```
:dvp ^ <SpeechAct>(confirm|accept|provide)
^ <Content>(#ctnt:content ^ <About>role ^ <GameNoun>#g)
^ <Context>(#ctxt:context ^ <Asker>robot ^ <CurrentGame>quiz)
^ <Control>(#ctrl:control ^ (<InsertApology>done|<InsertAck>extended))
^ <Action>(#act:)
=>
###a1 = rispondere ^ <Actor>(:person ^ 2nd ^ <Num>sg),
###a2 = domandare ^ <Actor>(:person ^ 1st ^ <Num>sg),
###a3 = fare ^ <Actor>(:person ^ 1st ^ <Num>sg) ^
        <Patient>(:abstract ^ #g ^ <Delimitation>unique ^ <Num>pl ^
        <Quantification>specific),
###a4 = ###a3: ^ <Recipient>(:m-benefactor ^ a ^ <Anchor>(:person ^ 2nd ^ <Num>sg)),

#act = :d-units ^ list ^
       <First>#act: ^
       <Next>(random(###a1,###a2,###a3,###a4):action ^ <Mood>ind ^ <Tense>pres),


// special case: confirm|accept|provide role
:dvp ^ <SpeechAct>(accept|provide)
^ <Content>(<About>role)
^ <Action>(#act: ^ <Next>(#n: ^ !<Mode>))
->
#n ^ <Mode>affirmative.

:dvp ^ <SpeechAct>confirm
^ <Content>(<About>role)
^ <Action>(#act: ^ <Next>(#n: ^ !<Mode>))
->
#n ^ <Mode>interrogative.
```

References

[1] Jason Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, school of Informatics, University of Edinburgh., 2002.

[2] Jason Baldridge and Geert-Jan M. Kruijff. *Coupling CCG and hybrid logic dependency semantics*. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003), Budapest, Hungary., 2003.

[3] Trevor Benjamin and Geert-Jan M. Kruijff. *MOLOKO: A CCG grammar for processing situated dialogue for HRI.*, 2008.

[4] Geert-Jan M. Kruijff Cem Bozsahin and Michael White. *Specifying Grammars for OpenCCG: A Rough Guide.*, 2008.

[5] Bernd Kiefer. *Content Planner Module for Talking Robots.*, 2014.

[6] Mark Steedman. *The syntactic Process*. The MIT Press, Cambridge Mass., 2000.

Ontologies and Reasoning for PAL

Hans-Ulrich Krieger

MLT@DFKI krieger@dfki.de

1 Ontologies for PAL

For the PAL project, the MLT@DFKI group has developed an ontology that consists of four sub-ontologies which are brought together via a set of interface axioms. The ontologies are encoded in OWL, being of species $\mathcal{SHIF}(\mathsf{D})$, according to the ontology metrics pane of Protégé v4.3.

1.1 Upper Ontology upp

The first ontology is a minimal and stripped-down upper ontology that we have originally developed for the EU projects MUSING, MONNET, and TRENDMINER, showing a tri-partite structure with the top-most classes upp:Abstract, upp:Happening, and upp:Physical. Most notable for PAL is the Happening representation which distinguishes between atomic upp:Situations and decomposable upp:Events, using properties such as upp:startsWith, upp:continuesWith, and upp:endsWith. This allows us to encode PDL-like processes and makes it also possible to define pre- and post-conditions.

1.2 Temporal Ontology time

The temporal ontology encodes the binary distinction between *synchronic* and *diachronic* properties through the (meta-)classes time:SynchronicProperty and time:DiachronicPropertry, subclasses of the class rdf:Property. This allows us to cross-classify properties as being open to a temporal change or staying constant. The time ontology also defines the property time:assigns in order to encode the continuation of a variable-like instances over time; see Section 2.1.

1.3 Combined DIT++ Dialogue & FrameNet Frame Ontology dafn

The most sophisticated ontology integrates ideas from the DIT++ taxonomy of dialogue acts (see `http://dit.uvt.nl`) and from the FrameNet lexical database (see `http://framenet.icsi.berkeley.edu/`). Frames and dialogue acts are modelled independently from one another, but dialogue acts incorporates frames through the property dafn:frame. Dialogue acts also encode the succession of dialogue acts through dafn:follows, define the dafn:sender and the dafn:addressee, but also allow a kind of embedding to model indirect speech in questions and answers via property dafn:refersTo. Modeling the shallow semantic arguments inside the frames frees us from defining repeating properties on various dialogue acts over and over again.

1.4 Domain Ontology dom

The fourth ontology represents knowledge about the PAL domain, relating classes through subclass axioms, e.g.,

dom:Child $\sqsubseteq$ dom:Animate $\sqsubseteq$ dom:Actor
dom:DiaryEntry $\sqsubseteq$ dom:Medium $\sqsubseteq$ dom:Activity
dom:Nursery $\sqsubseteq$ dom:Location
dom:Type1DM $\sqsubseteq$ dom:Diabetes $\sqsubseteq$ dom:Disease

The domain ontology also defines further XSD datatypes, such as xsd:kg, xsd:cm, xsd:mg_dL, xsd:mmol_L, etc. for domain-specific properties, such as dom:weight, dom:height, or, dom:bsl.

1.5 Brining it Together: The PAL Ontology pal

Importing and interlinking the four ontologies is achieved through OWL import statements and OWL axiom constructors, by incorporating domain/range restrictions, typing constraints, cardinality restrictions, and by directly relating classes, properties, and instances, for instance

dom:Actor $\sqsubseteq$ upp:Single
dom:Actor $\sqsubseteq$ dafn:Individual
dom:Location $\equiv$ upp:Place
dom:Activity $\sqsubseteq$ upp:Happening

Individuals that occur in a dialogue should usually be defined here, using namespace pal.

2 Reasoning and Querying with *HFC*

HFC is a bottom-up forward chainer and semantic repository implemented in Java which we have developed over the last years [5, 3, 4] and which is comparable to popular systems such as Jena [6] and OWLIM [2]. *HFC* supports RDFS and OWL reasoning à la [1] and [7], but at the same time provides a powerful language $\mathcal{RDL}$ for defining custom rules, involving functional *and* relational variables, complex tests and actions, and the replacement of triples in favor of tuples of arbitrary length. The inferential capabilities are complemented by a query language $\mathcal{QDL}$ which implements a relevant subset of SPARQL, but at the same time extends this language by adding, e.g., multi-valued aggregates.

2.1 Long-Term Storage of Dialogues

Dialogue-relevant information will either be encoded as time-stamped dialogue acts (e.g., Suggestion) or as time-stamped triples, leading to quadruples (quads). Events bear their time in a "container" object, the dialogue act (here: da42)

pal:da42 rdf:type dafn:Suggestion .
pal:da42 dafn:happens "583"^^xsd:long .

whereas triples are directly extended by a further temporal argument:

pal:lisa upp:location pal:nursery "488"^^xsd:long

Time stamps from a ticking clock, such as 583 and 488 above are represented as XSD long integers and can be seen as encoding *transaction time* (the time when a fact is entered to the data base), contrary to *valid time* (the time in which a fact is valid).

Other time-varying information in *HFC* that mimics the temporal continuation of an imperative variable in the *dialogue description language* $\mathcal{DDL}$ (e.g., lastDA) is also represented as a quad, using the time:assign property (see Section 1.2):

```
pal:lastDA time:assigns pal:da42 "583"^^xsd:long .
pal:lastDA time:assigns pal:da44 "622"^^xsd:long .
...
```

As we have already seen, the latter information is retrieved by a $\mathcal{DDL}$ program through a query, but this information is also "mirrored" back to the tuple store and might be accessed by rules as the dialogue progresses.

2.2 Rules, Queries, and Aggregates

At the moment, rules implement the standard RDFS [1] and OWL-Horst [7] entailment schemes. For instance (transitivity of the subclass relationship)

```
?c1 rdfs:subClassOf ?c2
?c2 rdfs:subClassOf ?c3
→
?c1 rdfs:subClassOf ?c3
```

Queries, involving aggregates, are posted to HFC through the dialogue description language $\mathcal{DDL}$. For instance, determining "my" (= i_myself) last outgoing dialogue act (lastDA; see above) in $\mathcal{DDL}$ is retrieved through the following $\mathcal{QDL}$ query:

```
SELECT ?da ?t
WHERE ?da rdf:type dafn:DialogueAct &
         ?da dafn:sender i_myself &
         ?da dafn:happens ?t
AGGREGATE ?dialact = LGetLatest ?da ?t "1"^^xsd:int
```

The aggregate LGetLatest here is responsible for returning the "latest" (XSD integer 1) dialogue act bound to ?dialact, given a table of two columns with headings ?da (for dialogue act) and ?t (for time), build up from the SELECT-WHERE part of the query.

3 Accepted Papers in 2015

– Hans-Ulrich Krieger & Thierry Declerck. *An OWL Ontology for Biographical Knowledge. Representing Time-Dependent Factual Knowledge*. Proceedings of the Conference on Biographical Data in a Digital World.

– Hans-Ulrich Krieger & Christian Willms. *Extending OWL Ontologies by Cartesian Types to Represent N-ary Relations in Natural Language*. Proceedings of the IWCS Workshop on Language and Ontologies.
– Hans-Ulrich Krieger & Stefan Schulz. *A Modal Representation of Graded Medical Statements*. Proceedings of the Conference on Formal Grammar.

References

1. Hayes, P.: RDF semantics. Tech. rep., W3C (2004)
2. Kiryakov, A., Ognyanov, D., Manov, D.: OWLIM – a pragmatic semantic repository for OWL. In: Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems. pp. 182–192 (2005)
3. Krieger, H.U.: A temporal extension of the Hayes/ter Horst entailment rules and an alternative to W3C's n-ary relations. In: Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS). pp. 323–336 (2012)
4. Krieger, H.U.: An efficient implementation of equivalence relations in OWL via rule and query rewriting. In: Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC). pp. 260–263 (2013)
5. Krieger, H.U., Kruijff, G.J.M.: Combining uncertainty and description logic rule-based reasoning in situation-aware robots. In: Proceedings of the AAAI 2011 Spring Symposium "Logical Formalizations of Commonsense Reasoning" (2011)
6. Reynolds, D.: Jena rules tutorial. In: Jena User Conference (2006), PowerPoint presentation
7. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3, 79–115 (2005)