The Role of Application Domain Knowledge in Using OWL DL Diagrams: A Study of Inference and Problem-Solving Tasks

Completed Research Paper

Joerg Leukel
University of Hohenheim
70599 Stuttgart, Germany
joerg.leukel@uni-hohenheim.de

Vijayan Sugumaran
Oakland University
Rochester, MI 48309, USA
sugumara@oakland.edu

Marvin Hubl
University of Hohenheim
70599 Stuttgart, Germany
marvin.hubl@uni-hohenheim.de

Abstract

Diagrammatic conceptual schemas are an important part of information systems analysis and design. For effectively communicating domain semantics, modeling grammars have been proposed to create highly expressive conceptual schemas. One such grammar is the Web Ontology Language (OWL), which relies upon description logics (DL) as a knowledge representation mechanism. While an OWL DL diagram can be useful for representing domain semantics in great detail, the formal semantics of OWL DL places a burden on diagram users. This research investigates how user's prior knowledge of the application domain impacts solving inference tasks as well as schema-based problem-solving tasks using OWL DL diagrams. Our empirical validation shows that application domain knowledge has no effect on inference performance but enhances schema-based problem-solving performance. We contribute to the conceptual modeling literature by studying task performance for a highly expressive modeling grammar and introducing inference tasks as a new task type.

Keywords: Conceptual modeling, IS analysis, Laboratory experiment, Task characteristics

Introduction

Understanding domain semantics is critical to stakeholders involved in information systems analysis and design (ISAD). For representing domain phenomena, diagrammatic conceptual schemas created using a modeling grammar are an important tool. One important characteristic of grammars is their expressiveness, i.e., the extent to which they provide constructs to represent relevant domain semantics (Gemino and Wand 2004; ter Hofstede and van der Weide 1993). A particular type of highly expressive conceptual schema is a diagram created using the Web Ontology Language (OWL) (W3C 2004), specifically its sublanguage *OWL DL*, with DL denoting description logics (Calvanese et al. 1998). While an OWL DL diagram represents the concepts and relationships within a domain, its modeling constructs embrace the formal semantics of description logics. The expressiveness of the OWL DL grammar has attracted some inquiry in information systems (IS) research (Bera et al. 2010; Sharman et al. 2004).

From an IS perspective, the role of OWL DL diagram is to assist analysts, developers, and users of information systems in understanding the domain of interest. Specifically, OWL DL diagrams have been proposed for communicating application domain semantics in the analysis phases of ISAD (Bera et al. 2010; Dermeval et al. 2015; Kaiya and Saeki 2006). Users who want to learn domain semantics from an OWL DL diagram must be aware of the formal semantics of modeling constructs used. Formal semantics enriches the semantics that is explicitly stated in the diagram by statements that are not visible but derived through logical inferences. Thus, users looking at a diagram must have some sense of the formal semantics to acquire a complete understanding of the domain semantics conveyed by the diagram.

Prior research has examined contextual factors that might affect how users understand a conceptual schema, including the user's application domain knowledge. While it was found that the role of application domain knowledge is contingent upon the task type (Khatri et al. 2006), the grammars used lacked formal semantics. Whereas OWL DL specialists are accustomed to interpreting the diagrams, other stakeholders face difficulties, which are due to the formal semantics (Rector et al. 2004; Warren et al. 2014). How far diagram users include formal semantics in their diagram understanding can be tested by *inference tasks*, which ask the user to infer statements from the diagram. While a computational reasoner could perform inferencing automatically, we see at least two scenarios in which using a reasoner might not always be possible or appropriate.

First, in the early analysis phases, OWL DL diagrams will often be interactively developed by a group of people involving domain experts, analysts, and modelers. These diagrams capture domain semantics at a rather high level of abstraction and will be revised during several rounds of discussions. Their development is made difficult by OWL DL modeling tools, which only provide limited visualizations for the large set of modeling constructs. For instance, many tools focus on few types of relationships between concepts but exclude equivalence, negation, union, and intersection. The latter constructs, however, are just those with formal semantics. Therefore, diagrams might be developed using other tools such as MS Visio, or no tool at all. In this case, automatic reasoners, which require an OWL serialization (file), cannot be employed for inferencing.

Second, for the setting described above, the use of UML-based representations of OWL DL has been proposed. In particular, some editors for UML class diagrams are being extended to include the constructs of OWL DL (Brockmans et al. 2004). This extension relies upon a so called UML profile for OWL (Djurić et al. 2005). Therefore, an OWL DL diagram will look very similar to a class diagram but may include additional symbols and labels to indicate the OWL-specific constructs. While the UML approach to OWL DL helps improve the modeling and visualization of OWL DL diagrams, reasoners still cannot be employed in those tools. Inferencing will only be possible if the resulting OWL file will be handed over to an OWL-specific tool. In summary, tool support is still inadequate because of separate tools being used for modeling and inferencing.

Considering the importance of formal semantics for understanding OWL DL diagrams, exploration of the factors that affect this understanding is still limited. We aim to fill this gap by investigating how the user's prior knowledge of the application domain impacts their task performance. In ISAD, team members are expected to have some basic knowledge of the application domain (Robillard 1999; Tiwana 2004). Sometimes, finding people with appropriate domain knowledge is difficult. In such circumstances, if the conceptual schemas contain enough domain semantics, then it may not be critical that the team members have some domain knowledge. If this can be ascertained with some degree of confidence through empirical evidence, then it makes it easy for project managers to form development teams since they don't have to be searching for people with enough domain knowledge. Hence, hypothesizing and empirically studying the impact of domain knowledge of the project team members on task performance is important.

Prior research informs how application domain knowledge affects task performance (Khatri et al. 2006) but no study has yet investigated OWL DL diagrams with formal semantics. Thus, the objective of our research is to empirically validate how application domain knowledge affects users of OWL DL diagrams in solving *inference tasks* as well as *schema-based problem-solving tasks*. While OWL DL diagrams provide all the information required to solve tasks of both types, knowledge of the application domain might have different effects. Similar to prior research, we base our argument on the theory of cognitive fit (Vessey 1991). The results of testing our hypotheses indicate that application domain knowledge has no effect on inference task performance but enhances schema-based problem-solving performance. Our research contributes to conceptual modeling by: (1) studying task performance for highly expressive conceptual schemas, and (2) introducing inference tasks as a new task type.

Our paper proceeds as follows. We first discuss the theoretical background for understanding OWL DL diagrams and present our hypotheses. Then, we report on the experiment we conducted to test these hypotheses. We discuss the findings from our study before concluding the paper.

Theoretical Background and Hypotheses

We refer to the *theory of cognitive fit* concerning how users employ representations for their problem-solving processes (Vessey 1991). This theory suggests that a match (fit) between the *problem representation* and the representation required for the *problem solving task* will enhance task performance. If there is a fit, the user can formulate a mental representation for task solution that uses corresponding information from both the problem representation and the problem solving task, which then improves task performance. In case of conceptual schema understanding, problem representations are the diagrams used and problem solving tasks are the types of *understanding tasks* that diagram users deal with. We first discuss OWL DL for problem representation. Then, we discuss understanding tasks and the role of application domain knowledge in solving these tasks to derive our hypotheses.

Problem Representation

An OWL DL schema can be represented in at least three different ways: diagram targeted at persons (OWL DL diagram), machine-readable format (e.g., XML), and abstract syntax of the underlying knowledge representation formalism. As with any modeling grammar, diagram understanding depends foremost on the ability to correctly interpret the constructs used in the diagram (syntax) and their mappings to concepts in the domain of interest (semantics) (Wand and Weber 2002). Because of description logics that underlies OWL DL, diagram users must also be aware of the impacts of that formalism on the diagram semantics. In discussing these impacts, we refer to the exemplar diagram shown in Figure 1.

While a variety of diagrammatic representations for OWL DL have been proposed (Katifori et al. 2007; Lanzenberger et al. 2010), the general approach is to use directed labeled graphs. Figure 1 illustrates the main constructs. *Concept* is the construct for representing domain phenomena at the type level (e.g., *Project*, *Person*), which can be linked by the *role* construct (e.g., *hasPerson*), and *individual* is the construct for capturing phenomena at the instance level (e.g., *William*).

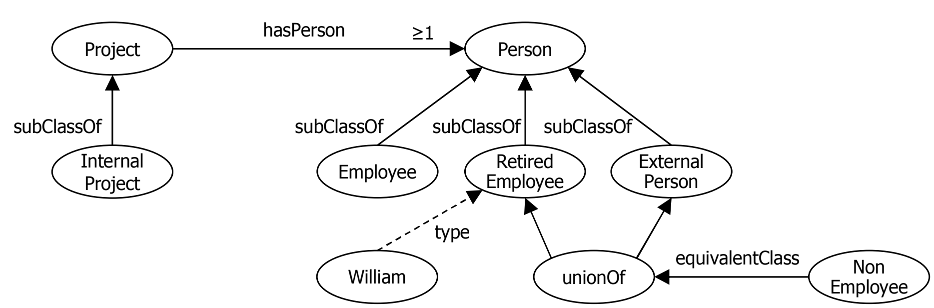


Figure 1. An excerpt of an OWL DL diagram for the project domain

Figure 1 contains some of the advanced modeling constructs, which are marked by lowercase labels on nodes (e.g., *unionOf*) and arrows (e.g., *type*). These labels have been standardized in the OWL specification. Unfortunately, the terminology defined in the OWL specification differs from the description logics literature (Baader et al. 2010). For instance, OWL uses *class* instead of *concept*, and *property* instead of *role*. Because of OWL's wide use in practice, visualizations implemented in modeling tools replace the DL-specific terms by the intuitive OWL terms (e.g., *subClassOf, equivalentClass*). In Appendix D, we provide a definition of the grammar used in both Figure 1 and our experiment.

The first characteristic of OWL DL is the larger set of constructs. Logical relationships between concepts other than the common IS-A-relationship can be described; these relationships include equivalence, negation, union, and intersection. For instance, the diagram shown in Figure 1 defines that *NonEmployee* is equivalent to the union of *RetiredEmployee* and *ExternalPerson* (represented by an arrow labelled *equivalentClass* from the *NonEmployee* node to the *unionOf* node, which then has un-labelled arrows to the two concept nodes). Figure 1 also provides examples of the IS-A-relationship, which is referred to as logical containment and represented by an arrow labelled *subClassOf* (e.g., *ExternalPerson* is contained in *Person*). The diagram shown in Figure 1 defines that the individual *William* (represented by a node) is an instance of the *RetiredEmployee* concept (represented by a dashed arrow with label *type*).

The second characteristic of OWL DL is the formal semantics of constructs. Formal semantics is the study of the meaning of a representation in terms of formal statements that are valid for that representation. If we create an OWL DL diagram, then its semantics are all the statements that are contained in the representation (explicitly stated) and all the statements that are also valid. The latter statements are implicit but can be concluded due to the underlying logic. For users of OWL DL diagrams, formal semantics is important to their understanding because the inferred statements are part of the diagram semantics. However, we stress that OWL DL inferences are different from *perceptual* inferences that users mentally construct when looking at a diagram. Perceptual inferences are due to intuition rather, and thus they are "extremely easy for humans" (Larkin and Simon, 1987, p. 98).

We illustrate formal semantics by inferring statements from the diagram shown in Figure 1, which defines that *William* is an instance of *RetiredEmployee*. Then, all statements about *RetiredEmployee* hold true for *William*. Hence, we can infer that *William* is an instance of (1) *Person* (because the *RetiredEmployee* concept is contained in the *Person* concept; represented by the *subClassOf* arrow between these two concepts), and (2) *NonEmployee* (because the *RetiredEmployee* concept is contained in the union that defines the *NonEmployee* concept; represented by the *unionOf* node and its *equivalentClass* arrow). Let us assume that the diagram contains two more individuals named *CloudProject*, which is an instance of *InternalProject*, and *Linda*, which has not been assigned to any concept. In addition, the two individuals are linked via the *hasPerson* role. We can conclude two statements: (1) *Linda* belongs to *Person* due to the *hasPerson* relationship with *InternalProject*, and (2) *CloudProject* belongs to *Project* because the *InternalProject* concept is contained in the *Project* concept.

This form of inferences is an important feature of OWL DL because it allows inferring some properties of an individual by its membership to a concept, without the need to directly assert these properties. This mechanism resembles the *categorization function of concepts* in human information processing (Smith and Medin 1981): If we know very little about an individual except that it belongs to a particular concept, then we can infer all or many of the concept's properties for that individual (considering all other statements). Furthermore, OWL DL allows us to draw inferences from an individual participating in a role (as discussed for the *Linda* example in the preceding paragraph). In this case, roles have a categorization function as well.

The formal semantics of OWL DL is subject to two assumptions, namely the *open-world assumption (OWA)* and the *non-unique name assumption (NUNA)*. Because neither assumption holds true for grammars such as the Entity-Relationship Model (ERM), newcomers to OWL DL will often face difficulties in integrating these assumptions into their problem-solving processes when using a particular schema (Rector et al. 2004). In the OWA, from the absence of a statement alone, it cannot be concluded that the statement is false. In other words, although our schema does not contain a statement, the statement might have been made somewhere else. For instance, the OWA impacts how to interpret the *RetiredEmployee* concept. The diagram states that *William* is an individual of *RetiredEmployee* but neither does the diagram define that *William* is not an instance of *Employee* nor can we infer such a statement.

In the *NUNA*, from two individuals having different names we cannot conclude that the two individuals are actually different; it could be the case that the two names designate the same individual in the domain. For instance, let us consider that *Paul* is an individual that has been assigned to *Employee* in Figure 1. From that statement alone, we cannot conclude that *Paul* and *William* are different. However, we could define *Employee* and *RetiredEmployee* as distinct concepts (syntax: arrow with label *complementOf*). Then follows that *William* is different from *Paul* due to both belonging to distinct concepts.

Understanding Tasks

Understanding tasks require the user to examine a diagram and derive information from that diagram. The information that the user derives from the diagram is his or her perception of domain semantics.

Understanding is best if the domain semantics articulated by the user equals the actual domain semantics; however, equivalence is difficult to achieve (Lindland et al. 1994).

The conceptual modeling literature distinguishes two types of understanding tasks (Khatri et al. 2006). First, *comprehension tasks* are concerned with domain elements that are represented in the diagram by modeling constructs. These tasks depend on a thorough understanding of the constructs used, and require mostly a surface-level understanding of the domain (Bodart et al. 2001). A deeper level of understanding is required by *problem-solving tasks* (which must not be confused with the general *problem solving task* construct in the theory of cognitive fit). A problem-solving task describes a problem in the domain and then asks the user to provide either explanations as to why this problem occurred or potential solutions to the problem. We focus on so called schema-based problem-solving tasks, which can correctly be solved by using information represented in the diagram (schema). Thus, the OWL DL diagrams provide all the information required for solving the task. Another type of problem-solving tasks require information beyond what is represented in the diagram; these tasks are referred to as inferential problem-solving task (Khatri et al. 2006). Next, we introduce inference tasks as a new task type and then discuss problem-solving tasks. Table 1 provides exemplars of the task types discussed.

Table 1. Task types used in conceptual modeling research

Task Type	Examplar with respect to Figure 1	Prior Studies
Comprehension task	Syntactic comprehension task: How many concepts are contained in the Person concept? a) 0, b) 1, c) 2, d) 3	Khatri et al. 2006
	Semantic comprehension task: A given employee must be assigned to: a) one project, b) more than one project, d) any number of projects, d) no project	Bodart et al. 2001; Khatri et al. 2006
Inferential problem-solving task	A project requires additional persons but no employees are available. What should the project leader do? Write as many alternatives as you can think of.	Bodart et al. 2001; Burton-Jones and Meso 2008; Gemino and Wand 2005
Schema-based problem-solving task	A quality manager asks for a report that identifies all the internal projects in which non-employees have been taken part. Based on the diagram provided, can you find an answer to the above problem.	Khatri et al. 2006
Inference task	Let us assume that Bill is equivalent to William. What statements about Bill can be inferred, if any?	-

Inference Tasks

Inference tasks are concerned with the formal semantics of modeling constructs used. Inference tasks ask the user to provide all the statements that can be inferred from a given statement. For instance, we can describe the inference task discussed earlier for Figure 1 as follows: "Let us assume that *Bill* is equivalent to *William*. What statements about *Bill* can be inferred, if any?" Diagram users can answer this question by trying to reproduce the steps of DL reasoning. This reproduction invokes deeper-level cognitive processes than comprehension tasks. Although users cannot avoid making perceptual inferences when looking at the diagram (Larkin and Simon 1987), they should focus on the logical inferences to maximize their task performance. Because the inferred statements are part of the diagram semantics, users must consider inferences when consulting an OWL DL diagram to learn domain semantics. Therefore, inference tasks are a useful vehicle for assessing how well the user has acquired an understanding of domain semantics conveyed by the diagram.

Our definition of inference tasks for OWL DL diagrams is consistent with the work of Larkin and Simon (1987) on how diagrammatic representations are processed. In their model, inference is the execution of some action "to add new (inferred) elements to the data structure" (p. 69). First, "data structure" can be considered as a mathematical graph that associates nodes via links. It is important to note that any diagram will be formalized using that structure. For instance, in case of OWL DL, the data structure would provide different types of nodes and links based on the grammar's constructs. Second, actions *(A)* will be triggered if some conditions *(C)* are fulfilled. While Larkin and Simon considered rules in the abstract form $A \rightarrow C$, the formal semantics of OWL DL is defined by such inference rules. Therefore, if a diagram user performs an inference task on a given OWL DL diagram, their knowledge of these inference rules will help in succeeding in the task. This task resembles the setting described by Larkin and Simon (p. 67): Although the OWL DL diagram is directly available to the user, the inference rules that operate on the diagram are in the user's memory.

Inference tasks for OWL DL diagrams as defined above must not be confused with other tasks that have been studied in conceptual modeling research. Specifically, the experiment reported by Dunn and Gerard (2001) had two sets of so called inference tasks, which were also motivated by the work of Larkin and Simon. The first set of tasks is concerned with how to derive database tables from an ERM diagram. Solving these tasks required knowledge of the derivation rules for the relational database model (RDM). Considering that inference adds elements to the data structure, in this case, the data structure is a composite structure that allows storing elements of both the ERM diagram and the relational database model. The major difference with our inference tasks is that the inference action processes elements of one modeling grammar (ERM diagram) to add elements using another modeling grammar (RDM). In addition, the inference rules are not due to the formal semantics of the constructs used in the diagram. The second set of tasks used by Dunn and Gerard (2001) asks if the ERM diagram allows making some specific statements within the application domain. Similarly, the answer, i.e., yes or no, cannot directly be stored using the data structure for ERM diagrams but requires a supplementary data structure. In summary, Larkin and Simon's concept of inference has much broader coverage of possible tasks than the type of inference used in our study, which relies only on the formal semantics of the grammar used.

Problem-Solving Tasks

Two types of problem-solving tasks have been identified (Khatri et al. 2006): *Schema-based problem-solving tasks* can be solved using information represented in the diagram, while *inferential problem-solving tasks* require using information beyond the information represented in the diagram. For the latter, making an inference means using tacit background knowledge of the domain, which is opposite to reproducing the DL reasoning as required for inference tasks. Thus, inferential problem-solving tasks must not be confused with inference tasks.

We focus on schema-based problem-solving tasks that come in the form of a query formulation task, which asks the user to describe whether and how some particular information can be retrieved from the diagram. Unlike a query against a database, the answer to this type of task will not be a statement in formal language but a textual description of how the user would use which elements of the diagram to solve the problem. Let us consider the following task directed at the diagram shown in Figure 1: *"A quality manager asks for a report that identifies all the internal projects in which non-employees have been taken part. Based on the diagram provided, can you find an answer to the above problem? If yes, describe how you would find the answer. Please be specific."* The solution to this task can be structured as follows: "First, retrieve all individuals of the *NonEmployee* concept. Second, select those that are linked via the *hasPerson* role with an individual of the *InternalProject* concept. Third, return the linked individuals of the *InternalProject* concept." Note that there might be other ways to describe the retrieval from the diagram, though any proposed solution can easily be validated against the diagram.

Role of Application Domain Knowledge

Application domain knowledge is the user's knowledge about the domain that is described in the diagram. If the task calls for implicit domain semantics that is not expressed in the diagram, then diagram users might bring their domain knowledge to bear in solving the task (Parsons and Cole 2005). However, under the assumption that a quality conceptual schema conveys *all* the relevant domain semantics, application domain knowledge will have little effect on diagram comprehension. The theoretical explanation is that comprehension tasks emphasize types of information that match closely to those in the diagram. This match of the types of information leads to cognitive fit, which facilities the mental processes and no transformation of the problem representation will be needed (Vessey 1991). Therefore, application domain knowledge is not essential to solving comprehension tasks, if the user thoroughly examines the diagram and interprets it correctly according to the syntax and semantics of the grammar used.

We regard inference tasks as a type of comprehension tasks. We contend that the mental representation for solving inference tasks can be formulated within the DL formalism and thus requires no transformation to the problem representation. The task representation begins with statements related to the diagram and then asks for the inferred statements. The problem representation, the OWL DL diagram, refers to the DL formalism based on a bijective mapping between the OWL labels of visual elements and the constructs of the underlying logic. Thus, the types of information emphasized in the inference task and in the diagram match, i.e., cognitive fit exists. This match provides strong motivation for the user to activate their general knowledge of DL reasoning and reproduce the steps of DL reasoning without bringing application domain knowledge to bear. Therefore, we posit the following hypothesis:

H1: *OWL DL diagram users with low application domain knowledge are equally accurate in inference tasks compared to those with high application domain knowledge.*

In case of schema-based problem-solving tasks, cognitive fit does not exist because the tasks emphasize other types of information than those in the diagram. Specifically, while the information emphasized in the problem representation are visual constructs with clearly defined formal semantics through the underlying logic, the information emphasized in the task does not match the OWL DL constructs and their formal semantics. Hence, the user must formulate a mental representation for task solution from two elements that emphasize different information, which makes the mental processes more intricate. Here, application domain knowledge might help in the processes for formulating the mental representation and ultimately improve task performance. Application domain knowledge will facilitate the transformation of different information types required for problem solving. This effect of application domain knowledge received empirical support for users of diagrams produced from two variants of the ERM grammar (Khatri et al. 2006). We posit that this effect also holds true for users of OWL DL diagrams and state the following hypothesis:

H2: *OWL DL diagram users with high application domain knowledge are more accurate in schema-based problem-solving tasks compared to those with low application domain knowledge.*

Method

To test our hypotheses, we conducted a controlled laboratory experiment for which we report the experimental design and data collection in this section.

Experimental Design

Participants were given two OWL DL diagrams using which they had to perform inference and schema-based problem-solving tasks. Both diagrams were syntactically equivalent except for the labels used to describe the application domain. The factor under investigation was *application domain knowledge* and had two levels, *high* vis-á-vis *low*. We used a repeated measures design, where each participant was exposed to both treatment conditions. This design allowed us to effectively control for individual differences such as knowledge of the modeling grammar. Participants were randomly assigned to either the group that first started with the diagram for the familiar domain or the group that started with the diagram for the unfamiliar domain. For each diagram, inference tasks were followed by the schema-based problem-solving tasks. We chose this order because the schema-based problem-solving tasks required the participants to examine the entire complex diagram; this examination can be facilitated by first solving less demanding tasks (Khatri et al. 2006). Within each task type, the tasks were presented in two different sequences (because of two groups, there were in total four different sequences of tasks).

Participants

The experiment was targeted at persons who have practical experience with the OWL DL grammar. The participants should have an understanding of the modeling constructs including their formal semantics and be able to retrieve domain semantics from diagrams. Therefore, the target population for the study was novice analysts in ISAD who received academic training in conceptual modeling and possess some experience with the OWL DL grammar.

To determine the required sample size, we performed an a priori power analysis using the *G*Power 3.1* tool (Faul et al. 2007). In this analysis, we assumed a medium size effect of application domain knowledge on schema-based problem-solving performance and set d=0.60. We selected the Wilcoxon-signed ranked test (one-tailed). The analysis revealed that we must have at least 20 participants providing 40 observations to achieve a sufficient statistical power (*P*=.8). Therefore, we recruited 26 students for the experiment (providing 52 observations). This sample size was large enough to detect medium size effects for our repeated measures design.

Our participants were undergraduate IS students (24 males and 2 females) enrolled in a *Knowledge Engineering* course. Participation was voluntary but participants were awarded three extra credit points for the final exam (total of 60 points). Because this was a compulsory course in the sixth semester of study, the students were familiar with conceptual modeling and modeling grammars. The course provided an introduction to the OWL DL grammar through three classroom sessions of each 90 minutes in the weeks prior to the experiment as follows: The first session introduced DL in the form of axiomatic and visual grammar. In the second week's session, students had to design an OWL DL diagram from a textual description and derive axioms from a given diagram. The third session was dedicated to the reasoning capabilities of OWL DL by discussing nine exemplar diagrams and, finally, inferring statements from a larger diagram, which was of similar size as the diagrams used in the experiment. In summary, the students' background suggests that the participants can serve as surrogates for novice business analysts and modelers who are involved in the analysis phase of ISAD (based on the training provided in the class).

Measurements

Our experiment had two dependent variables for measuring *inference performance* and *schema-based problem-solving performance*. The inference tasks required to write down statements inferred from the diagram. We used "negative marking" for the inference tasks by penalizing the wrong answers. For instance, the correct solution to the first inference task (provided in Appendix B) comprises three statements, with each getting a score of one-third. If the answer contained two correct statements and one incorrect statement, then we would assign a score of only one-third to the answer. The assessment was straightforward by comparing the answer with the correct solution. In addition, we verified the correctness of the solutions that we used by (1) implementing each diagram with *Protégé 4.3* editor (Protégé 2016), and (2) retrieving the inferred statements from its built-in DL reasoner.

Responses to the schema-based problem-solving tasks were independently coded by four student assistants, who were familiar with interpreting conceptual schemas but were unaware of our hypotheses under investigation. The correct solution to each task included two elements, each of which received a score of one. Thus, the maximum score that can be achieved for each domain was six (for the three tasks).

We ensured the validity of the coded data through the following procedure: First, we developed a coding manual, which included correct solutions and their coding as well as exemplars of incorrect solutions. For the familiar domain diagram, we provided an example coding instruction and discussed its use with the student assistants. The actual coding started for the first five answers to task 1 (familiar domain). Next, the four coders compared and discussed their ratings to resolve any inconsistency (no intervention by the researchers). Afterwards, the group discussed their results with one of the researchers who reviewed the group ratings. Then, the remaining 21 responses were coded, followed by the two rounds of discussion. This procedure was repeated for all of the six tasks (for the final two tasks, the coders were given all 26 answers at once because of already high consistency). Considering that we used four independent coders, their raw agreement was high with 86.5% for the familiar domain and 85.9% for the unfamiliar domain. In very few cases, the discussion between the group of coders and the researcher led to a revision of the code assigned (this occurred for just 6 of the 2*6*26=312 responses).

Materials

The materials included four parts and a supplement. The first part captured the participants' demographic background (materials provided in Appendix A). To determine how far along the students are in their degree program, we asked for their undergraduate credits and the grade in the *Modeling* course (which provides an introduction to ERM and the *Business Process Model and Notation* (BPMN) but not OWL DL). With respect to self-reported modeling knowledge, we adopted a three-item instrument from Mendling et al. (2010), and adjusted the items to the OWL DL grammar. While we had confidence in our selection of the familiar and unfamiliar domains (see below), we also assessed self-reported domain knowledge (through one item each).

The second part of our materials provided a tutorial on how to solve inference and schema-based problem-solving tasks. While both task types were known to our participants through attending the classroom sessions, we included this tutorial to refresh their knowledge. First, participants independently worked on the tutorial tasks. Then, the experimenter presented the correct solutions. Thus, the participants received feedback on their solution. The tutorial diagram represented a very familiar domain, namely *IS program*, which described concepts such as *Course*, *Teacher*, and *Student*. This diagram was smaller than those used in the treatment phase but made use of the same grammar constructs. We first asked the participants to perform three syntactic comprehension tasks, followed by two inference tasks, and, finally, one schema-based problem-solving task.

The third and fourth part of our materials provided the diagrams and tasks (based on which treatment condition was administered first). The two diagrams were syntactically equivalent (number of elements, diagram layout) except for the domain labels used. We first developed the diagram representing an application domain of high familiarity (Figure 2). Because our participants were IS students, we chose *project management* as a familiar domain; it provides concepts such as project task, subtask, person, and qualification. The students' background knowledge can interfere with the diagram semantics; hence, this domain was appropriate for our investigation. Then, we developed a diagram representing *road surface laying* (Figure 3). We chose this domain because we expected that IS students would be more familiar with project management than with road surface laying, which is a subject studied in civil engineering. The diagram was developed by the third author who has worked two and a half years in civil engineering research projects and closely collaborated with domain experts. The diagram represents knowledge described in a reference document published by the German Asphalt Pavement Association (2011). An alternative to choosing such a real domain would have been an abstract domain for which the diagram only includes "void" labels such as *Alpha*, *Beta* and so on. While this choice would have increased the contrast between the levels of our independent variable, it is unlikely in practice that diagram users will have absolutely no knowledge of the domains represented in the diagrams they use (Bera et al. 2014).

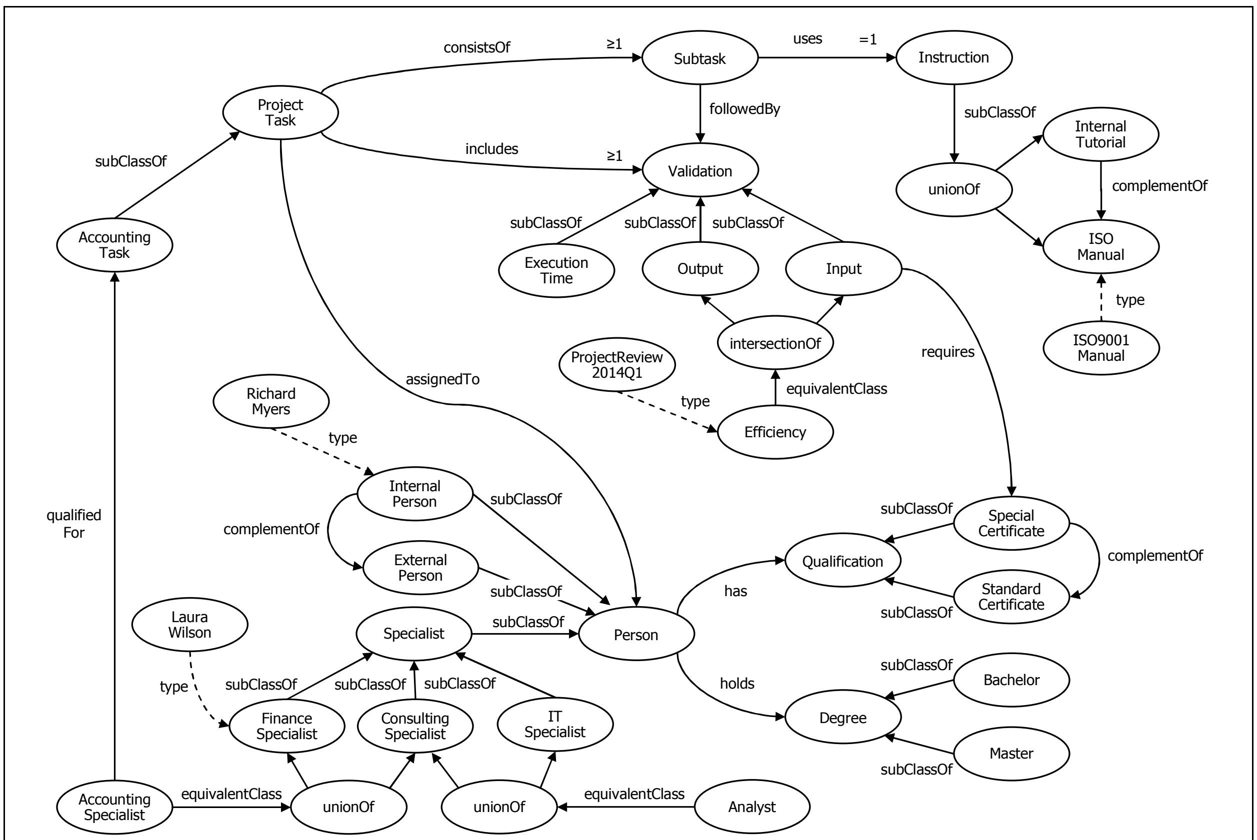


Figure 2. OWL DL diagram for the familiar domain

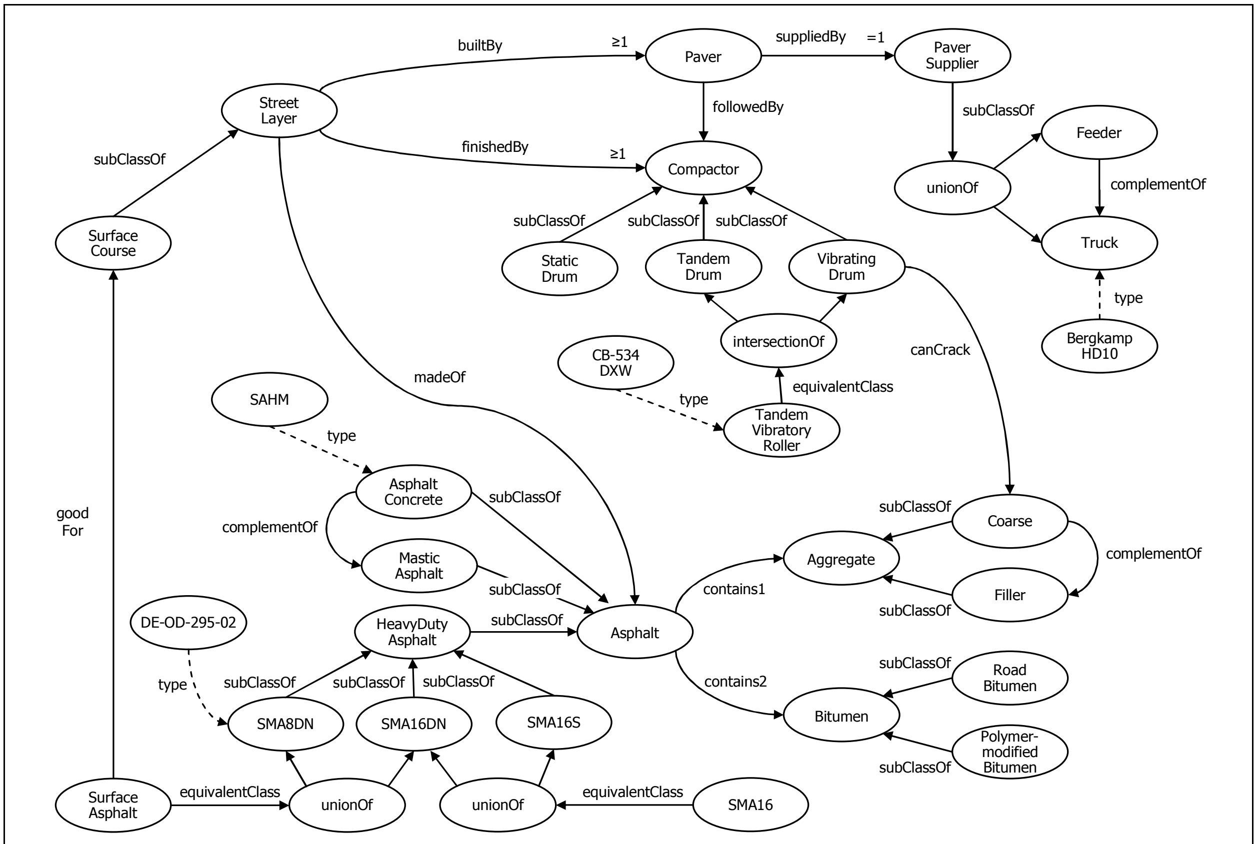


Figure 3. OWL DL diagram for the unfamiliar domain

In developing the inference tasks (provided in Appendix B), we ensured that the inferred statements to be derived by the users were based on the formal semantics of constructs not included in the ERM grammar but specific to OWL DL. The correct solution to all tasks included 13 inferred statements per diagram, which were based on the following constructs: Role (2), concept inclusion (4), concept equivalence (2), concept negation (2), concept union (1), and concept intersection (2). The number of statements to be inferred per task ranged from one to three. While four tasks asked for inferred statements about a particular individual already contained in the diagram, two tasks began with some information about a new individual and then asked for the inferred statements (tasks 3 and 6).

With respect to the schema-based problem-solving tasks (provided in Appendix C), we adopted the format of query formulation tasks used in prior research (Khatri et al. 2006). Each task required the user to analyze several elements of the diagram that were linked via roles and logical relationships, respectively. Participants were asked how to solve the problem by identifying the relevant elements and their links.

The supplement provided during the experiment included the modeling grammar (Appendix D).

Procedures

The experiment was organized as a class room exercise. Participants were informed that the experiment would provide an assessment of their learning progress; the research objective and the treatment conditions exposed to were not disclosed. After two researchers explained the procedures and answered questions, participants were randomly assigned to either the familiar or unfamiliar domain group. Then, the first and second parts of the materials were distributed, which included the questions on the participants' background and the tutorial, respectively. After all participants had completed and returned the materials, one instructor presented the correct solutions to the tutorial tasks. Then, the first treatment phase started and distributing the respective paper-based treatment materials. One seat was left empty between every two participants, who were, in addition, assigned to different groups. The instructors made sure that there was no collaboration between participants. The participants were given ample time to work through the inference tasks followed by the schema-based problem-solving tasks. Once a participant had completed the tasks, an instructor collected the material and distributed the material for the second treatment condition. The *time required* for each treatment phase was recorded by the instructors, and considered as a control variable in the data analysis.

Results

This section reports the results by first examining the conformance of the data with the assumptions of statistical tests. Then, we present the results from testing our hypotheses.

Data Screening

Table 2 shows participants' demographic data and controls. Participants' study progress ranged between completion of the third semester and beginning of the sixth semester with a mean of fourth semester (one student did not report credits). All students had passed the *Modeling* course and reported fairly good grades (three students reported passing but not their grade). On average, participants rated their DL modeling knowledge as in the middle. They also indicated that they possess more knowledge of project management (familiar domain) than road surface laying (unfamiliar domain); a Wilcoxon signed-rank test showed that the difference was significant (Z=-4.198, p<.001). Our participants performed very well in the tutorial's syntactic comprehenion tasks (relative score of 95%). The scores for the inference and schema-based problem-solving tasks were, as expected, lower (45% and 27%, respectively). Overall, the data suggests that our participants had knowledge of the grammar as required for our investigation.

Table 2. Participants' data and controls (N=26)

Variable	Scale	Min	Max	M	Mdn	SD
Age	Years	20	34	22.38	22.00	2.52
Undergraduate credits	0-180	90	160	142.76	147.00	16.88
Grade in the *Modeling* course	1-5[1]	1.3	3.3	2.36	2.30	0.58
Self-reported knowledge: DL modeling	1-7	2.0	6.33	4.15	4.33	1.19
Self-reported knowledge: Familiar domain	1-7	2	6	4.62	5.00	0.98
Self-reported knowledge: Unfamiliar domain	1-7	1	6	2.35	2.00	1.36
Tutorial: Syntactic comprehension	0-3	2	3	2.85	3.00	0.37
Tutorial: Inference	0-2	0	2	0.90	0.80	0.64
Tutorial: Schema-based problem-solving	0-3	0	3	0.81	0.00	1.20
Time required: Familiar domain	Mins.	14	25	19.31	20.00	3.26
Time required: Unfamiliar domain	Mins.	15	25	20.38	20.50	3.21

[1] Inverse scale with lower values indicating higher levels of knowledge.

With respect to the reliability of our multi-item instruments, we found sufficient levels of internal consistency as indicated by Cronbach's alpha (.895 for self-reported modeling knowledge, .861 for inference performance/familiar domain, .775 for inference performance/unfamiliar domain). We examined all dependent variables for normal distribution and found deviations so that we decided to run non-parametric tests for our hypotheses testing.

Prior to hypotheses testing, we inspected the correlations between our independent and dependent variables. First, there were no significant correlations between any self-reported variable (i.e., age, credits, grade, DL modeling knowledge, application domain knowledge) and inference performance. Second, there was only one significant correlation with schema-based problem-solving performance (between knowledge of the unfamiliar domain and performance in that domain with r=-.391). Third, there were no significant correlations between the time required and any of the performance variables.

Test of Hypotheses

Table 3 provides the results of our hypotheses testing. With respect to inference performance (H1), we found no significant difference in using the OWL DL diagram for the familiar domain compared to using the diagram for the unfamiliar domain. Power analysis using the *G*Power 3.1* tool revealed that in order for an effect of this marginal size (d=0.0284) to be detected (80% chance) as significant at the 5% level, a sample of 8,024 participants would be required. Thus, it is unlikely that our nonsignificant finding can be attributed to a limited sample size. With respect to schema-based problem-solving performance (H2), participants using the diagram for the familiar domain achieved higher scores than when using the diagram for the unfamiliar domain, and the size of this effect was medium.

Table 3. Results

Variable	Scale	Application Domain						Test Results		
		Familiar			Unfamiliar					
		M	Mdn	SD	M	Mdn	SD	Z	p[1]	Effect size[2]
H1: Inference performance	0-6	3.38	3.58	2.09	3.41	3.67	1.82	-0.105	.916	n/a
H2: Schema-based problem-solving performance	0-6	2.77	3.00	1.48	1.62	2.00	1.42	-3.068	.002	Medium (0.43)

[1] Significant at p<.05 (Wilcoxon signed-rank test, asymptotic, 2-tailed).

[2] Effect size: absolute r; >0.1 Small, >0.3 Medium, >0.5 Large (Cohen 1988).

Because of our repeated measures design, learning might have confounded the observed effects. Therefore, we additionally formed two groups of observations and compared their results. The first group contained all observations for the first experimental run (either familiar or unfamiliar domain), while the second group contained those for the second experimental run (again, either familiar or unfamiliar domain). Then, we ran Mann-Whitney U-tests for our dependent variables and the time required. The differences for both inference and schema-based problem-solving performance were nonsignificant. However, we found that participants spent less time in the second run compared to the first run. On average, the decrease was 5.08 minutes for the familiar domain and 4.77 minutes for the unfamiliar domain (p<.001). In summary, our results suggest that while learning led to faster task completion it did not affect task performance.

Discussion

We discuss the contributions, implications, and limitations of our research.

Contributions to IS Literature

Investigating the context in which conceptual schemas are created and used is an important part of conceptual modeling research (Wand and Weber 2002). Our research examined two contextual factors, the user's application domain knowledge as an *individual difference factor* and schema understanding as a *task factor*. Specifically, the research question addressed was how application domain affects the performance of users of OWL DL diagrams in solving inference tasks and schema-based problem-solving tasks. Based on our empirical results, our research makes the following three specific contributions to IS research.

First, our study contributes to the literature by investigating schemas that are highly-expressive because of the knowledge representation formalism that underlies the OWL DL grammar. The expressiveness of modeling grammars is a major subject of conceptual modeling research. Of particular importance is the notion of *ontological expressiveness*, which measures whether and how grammatical constructs map onto the constructs of a philosophical ontology (Wand and Weber 1993). Since ontology as a branch of philosophy provides theories about the structure and behavior of the world in general, they can be useful for evaluating grammars (e.g., Bera et al. 2014; Parsons 2011; Shanks et al. 2008). Prior IS research has focused on ontological expressiveness, whereas *inferential expressiveness* due to description logics has yet received little attention. Because OWL DL and the grammars studied in prior research have different origins in logic, the contextual factors when using OWL DL diagrams should be investigated.

Second, empirical research on the understandability of OWL DL diagrams is still scarce and has only considered alternative visualizations of grammar constructs (Fu et al. 2014; Garcia-Penalvo et al. 2012; Motta et al. 2011). Our study fills this gap in the literature by investigating two contextual factors through a rigorous empirical evaluation. We extend prior research that has used the theory of cognitive fit to explain and predict task performance for ERM diagrams in the presence of application domain knowledge (Khatri et al. 2006). Our study is the first empirical investigation of performing inference tasks. Prior research has studied only the common constructs in the ERM grammar (e.g., Khatri et al. 2006; Parsons 2011; Poels et al. 2011) but not the constructs that enable DL-based inference tasks. We note that diagram users must be aware of the formal semantics of constructs used, and the inferred statements that are part of the domain semantics conveyed by a diagram. Understanding *inferences* is important when diagrams are used in the analysis phase where tool-based reasoners will not be available or their use is not appropriate. Our study provides evidence that cognitive fit between the information emphasized in the OWL DL diagram and the information required for solving inference tasks indeed exists. Cognitive fit is possible because of a bijective mapping between the OWL labels of visual elements and the constructs of the underlying logic, which were known to the participants of our experiment. We also find that application domain knowledge helps the user in schema-based problem-solving tasks because the problem representation in the OWL DL diagram and the representation required for solving the task do not match. We believe that these findings can be useful for ISAD project managers in deciding about the required expertise of project teams that use OWL DL diagrams.

Third, our research helps clarify the notion of inferencing used in the conceptual modeling literature by referring to formal semantics and investigating a particular knowledge representation mechanism that is well understood in the knowledge engineering literature (Baader et al., 2010). Prior research assigned inferencing to very different tasks including the derivation of statements in one grammar from statement expressed in another grammar (Dunn and Gerard 2001) and inferential problem-solving tasks that require domain knowledge beyond what is available from the diagram (Bodart et al. 2001; Burton-Jones and Meso 2008; Gemino and Wand 2005).

Implications

Our study results have several implications for future research. For research streams investigating schema understanding, our study adds to the literature by introducing inference tasks as a new task type. We focused on two groups of OWL DL constructs that enable inferences, i.e., logical relationships between concepts (inclusion, equivalence, union, and intersection) as well as roles. OWL DL provides further constructs such as role restrictions and different types of roles, which also have formal semantics beyond that of ERM or UML class diagrams. Future research could investigate whether our empirical support for cognitive fit of inference tasks holds for all the constructs. While we defined inference tasks that required the user to interpret several constructs used in the diagram, our instrument for measuring inference performance was reliable. However, we did not hypothesize about specific constructs.

Second, opportunities exist to further explain the cognitive processes for interpreting constructs defined in the OWL DL grammar. While we studied how users interpret diagrammatic representations conforming to an underlying knowledge representation mechanism, past research informs how humans perform logical deductions from formal statements (e.g., Rips 1983; Johnson-Laird and Byrne 1991). Similarly, description logics researchers have begun using theories of mental models to study how well readers interpret formal non-diagrammatic DL statements. For instance, in the experiment reported by Warren et al. (2014), participants were given a set of statements and a proposed inference. Then, the participants had to decide whether the proposed inference was valid or not. While this type of inference task is different from our definition, approaching the problem from different theoretical perspectives could extend our understanding of human information processing for highly expressive conceptual schemas.

Third, other researchers could extend the design of our study with respect to the individual difference factor of conceptual modeling. We considered the user's prior knowledge of the application domain and found that this knowledge affects schema-based problem-solving performance. This finding is important because the effect observed is very similar to the role of application domain in schema-based problem-solving using ERM diagrams (Khatri et al. 2006). Our results suggest that diagram users, who received training in the formal semantics of OWL DL, still bring domain knowledge to bear in solving query formulation tasks. It would be interesting to know whether users with little or no knowledge of the OWL DL grammar would also be able to solve such tasks due to their prior domain knowledge. In a similar vain, fellow researchers could use our definition of inference tasks to study how the user's knowledge of and practical experience with the OWL DL grammar impacts their inference performance. While our participants received academic training on OWL DL, research could investigate how inference tasks are approached by participants with knowledge of the ER grammar but no knowledge of OWL DL. This design would mirror a setting found in practice where naïve users analyze OWL DL diagrams.

Limitations

The limitations of our study must be noted. First, limitations exist due to using students as surrogates for novice analysts that validate domain semantics conveyed by a diagram. Our students lacked the in-depth experience with OWL DL that professional analysts would possess. While we provided them training on the formal semantics of OWL, including the DL-specific assumptions (i.e., OWA and NUNA), this setting limits the external validity of our results. Second, while we had planned to use an existing schema that has been created and validated in prior research, our search in the OWL repositories listed on the W3C website was unsatisfactory. The schemas were either too specific with respect to the application domain, or lacked the use of the advanced modeling constructs required by our hypotheses, or had both deficiencies. Third, our tasks were limited in number and complexity, and we used only one diagram for each treatment condition. This setting is similar to prior studies on diagram comprehension (e.g., Agarwal et al. 1999; Khatri et al. 2006; Poels et al. 2011). Fourth, we chose a subset of the OWL DL specification, which provides more modeling constructs. The effects observed for the constructs used in our experiment cannot necessarily be generalized to all the advanced constructs.

Conclusion

OWL DL is a conceptual modeling grammar characterized by the larger set of modeling constructs to describe domain constraints, and the formal semantics of the constructs based on an underlying knowledge representation mechanism. These characteristics make it necessary to explore the factors that affect how users of OWL DL diagrams understand domain semantics. We find that the user's application domain knowledge has no effect on inference performance but enhances schema-based problem-solving performance. We contribute to the conceptual modeling literature by conducting the first empirical study on task performance for the highly expressive OWL DL grammar, which adds description logics-based inference tasks.

Acknowledgements

The work of J. Leukel has been partially supported by the Federal Ministry of Education and Research, Germany, under grant 16KIS0083. The work of V. Sugumaran has been supported in part by a 2016 School of Business Administration Spring/Summer Research Fellowship from Oakland University.

Appendices: Experimental Materials

Appendix A: Participant Data

Demographics

Age (years); Gender (male/female); Undergraduate credits (max. 180); Modeling course: attendance (yes/no), passed exam (yes/no), grade.

Self-reported modeling knowledge (7-point scale from "strongly disagree" to "strongly agree")

- "Overall, I am very familiar with description logics."
- "I feel very confident in *understanding* knowledge bases created with description logics."
- "I feel very competent in *using* description logics for conceptual modeling."

Self-reported application domain knowledge (7-point scale from "very low" to "very high")

- "To what extent do you have knowledge of *project management procedures*?"
- "To what extent do you have knowledge of *road surface laying procedures*?"

Appendix B: Inference Tasks

Note: To conserve space, we present only the inference tasks for the familiar domain. In the inference tasks for the unfamiliar domain, all terms printed in bold were replaced by the corresponding terms as shown in Figure 3.

Task	**Solution**
List all statements about **ProjectReview2014Q1** that can be inferred, if any.	Instance of: *Output*, *Input*, *Validation*. (one-third per statement)
List all statements about **ISO9001Manual** that can be inferred, if any.	No instance of *InternalTutorial*.
Let us assume that **ProjectReview2014Q1** is connected via the **requires** role with **SigmaPlus**. List all statements about **SigmaPlus** that can be inferred, if any.	Instance of: *SpecialCertificate*, *Qualification*; no instance of *StandardCertificate*. (one-third per statement)
List all statements about **LauraWilson** that can be inferred, if any.	Instance of: *Specialist*, *AccountingSpecialist*, *Person*. (one-third per statement)
List all statements about **RichardMyers** that can be inferred, if any.	Instance of *Person*; no instance of *ExternalPerson*. (one-half per statement)
Let us assume that **BUpdate2014** is connected via the **assignedTo** role with **LauraWilson**. List all statements about **BUpdate2014** that can be inferred, if any.	Instance of *ProjectTask*.

Appendix C: Schema-Based Problem-Solving Tasks

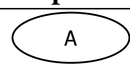
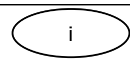
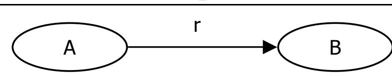
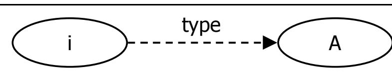
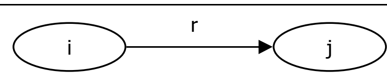
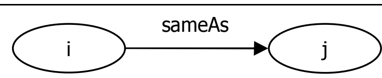
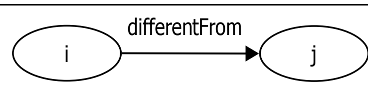
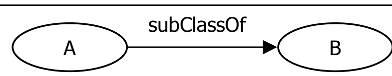
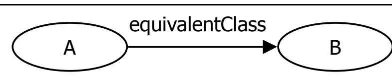
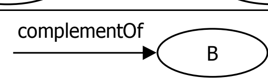
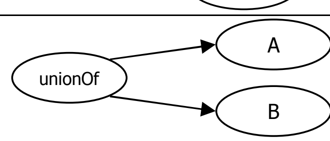
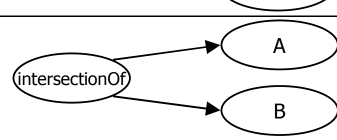
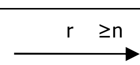
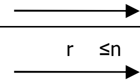
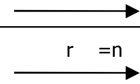
Familiar Domain

Task	**Solution**
A human resources manager needs to review a completed project task. This review is based on the profile of persons involved in the task.	1. Retrieve individuals of the "Person" concept linked via the "assignedTo" role with the "ProjectTask" concept. 2. Retrieve information about each person such as degrees, qualifications, and types of persons.
A business process analyst wants to know all the tasks to be carried out and the persons involved in a project.	1. For each individual of the "ProjectTask" concept, retrieve the subtasks and validation tasks via the "consistsOf" and "includes" roles. 2. Via the "assignedTo" role, retrieve individuals of the "Person" concept.
A professional trainer asks for a report that identifies all the ISO manuals that have been used for tasks by holders of master's degrees.	1. Retrieve individuals of the "Person" concept that are linked via the "holds" role with an individual of the "Master" concept. 2. For these individuals, select via the roles "assignedTo" – "consistsOf" – "uses" the individuals of the "ISOManual" concept.

Unfamiliar Domain

Task	**Solution**
A surveyor needs to write a specification as part of their street layer analysis. This specification is based on the physical composition of the surveyed object.	1. Retrieve individuals of the "Asphalt" concept linked via the "madeOf" role with the "StreetLayer" concept. 2. Retrieve information about each asphalt such as aggregates, bitumens, and types of asphalts.
A civil engineer wants to know all the activities to be carried out and the materials involved in road surface laying.	1. For each individual of the "StreetLayer" concept, retrieve the paving and compacting activities via the "builtyBy" and "finishedBy" roles. 2. Via the "madeOf" role, retrieve individuals of the "Asphalt" concept.
The Environmental Protection Agency wants to create a report that identifies all the trucks that have been exposed to polymer-modified bitumen.	1. Retrieve individuals of the "Asphalt" concept that are linked via the "contains2" role with an individual of the "Polymer-modified Bitumen" concept. 2. For these individuals, select via the roles "madeOf" – "builtBy" – "suppliedBy" the individuals of the "Truck" concept.

Appendix D: Modeling Grammar

Construct	**DL Syntax**	**Visual Representation**	**Definition**
Concept	A		Represents a set of individuals.
Individual	{i}		Represents an individual.
Role	(A, B):r		Represents a relation (role) between two concepts.
Concept assignment	{i}:A		Assigns individual i to concept A (i is an instance of A).
Role assignment	({i}, {j}):r		Represents a relation between two individuals.
Equivalent individuals	{i} ≡ {j}		Individuals i and j are equivalent.
Distinct individuals	{i} ⊑¬{j}		Individuals i and j are different from each other (distinct).
Concept inclusion	A ⊑B		Concept A is contained in concept B (or: B includes A).
Concept equivalence	A ≡ B		Concepts A and B are equivalent.
Concept negation	¬B		Represents the negation of a concept.
Concept union	A ⊔B		Represents the union of two or more concepts.
Concept intersection	A ⊓B		Represents the intersection of two or more concepts.
Minimum cardinality	≥n r		The minimum cardinality of r is n.
Maximum cardinality	≤n r		The maximum cardinality of r is n.
Exact cardinality	=n r		The exact cardinality of r is n.

References

Agarwal, R., De, P., and Sinha, A. P. 1999. "Comprehending Object and Process Models: An Empirical Study," *IEEE Transactions on Software Engineering* (25:4), pp. 541-556.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F. (eds.). 2010. *The Description Logic Handbook: Theory, Implementation, and Applications* (2nd ed.). New York: Cambridge University Press.

Bera, P., Burton-Jones, A., and Wand, Y. 2014. "Research Note–How Semantics and Pragmatics Interact in Understanding Conceptual Models," *Information Systems Research* (25:2), pp. 401-419.

Bera, P., Krasnoperova, A., and Wand, Y. 2010. "Using Ontology Languages for Conceptual Modeling," *Journal of Database Management* (21:1), pp. 1-28.

Bodart, F., Patel, A., Sim, M., and Weber, R. 2001. "Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests," *Information Systems Research* (12:4), pp. 384-405.

Brockmans, S., Volz, R., Eberhart, A., and Löffler, P. 2004. "Visual Modeling of OWL DL Ontologies Using UML," in *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, S. A. McIlraith, D. Plexousakis, and F. van Harmelen (eds.), Berlin: Springer, pp. 198-213.

Burton-Jones, A., and Meso, P. 2008. "The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model," *Journal of the Association for Information Systems* (9:2), pp. 748-802.

Calvanese, D., M. Lenzerini, and Nardi, D. 1998. "Description Logics for Conceptual Data Modeling," in *Logics for Databases and Information Systems*, J. Chomicki, and G. Saake (eds.), New York: Springer US, pp. 229-263.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.), Hillsdale, NJ: Lawrence Erlbaum Associates.

Dermeval, D., Vilela, J., Bittencourt, I. I., Castro, J., Isotani, S., Brito, P., and Silva, A. 2015. "Applications of Ontologies in Requirements Engineering: A Systematic Review of the Literature," *Requirements Engineering*. Advance online publication. doi: 10.1007/s00766-016-0258-2.

Djurić, D., Gašević, D., Devedžić, V., and Damjanović, V. 2005. "A UML Profile for OWL Ontologies," in *Model Driven Architecture*, U. Aßmann, M. Aksit, and A. Rensink (eds.), Berlin: Springer, pp. 204-219.

Dunn, C. L., and Gerard, G. J. 2001. "Auditor Efficiency and Effectiveness with Diagrammatic and Linguistic Conceptual Model Representations," *International Journal of Accounting Information Systems* (2:4), pp. 223-248.

Faul, F., Erdfelder, E., Lang, A. G., and Buchner, A. 2007. "G* Power 3: A Flexible Statistical Power Analysis Program for the Social, Behavioral, and Biomedical Sciences," *Behavior Research Methods* (39:2), pp. 175-191.

Fu, B., Noy, N. F., and Storey, M.-A. 2014. "Eye Tracking the User Experience – An Evaluation of Ontology Visualization Techniques," *Semantic Web*, Preprint (online at http://content.iospress.com/articles/semantic-web/sw163; retrieved April 24, 2016).

García-Peñalvo, F. J., Colomo-Palacios, R., García, J., and Therón, R. 2012. "Towards an Ontology Modeling Tool. A Validation in Software Engineering Scenarios," *Expert Systems with Applications* (39:13), pp. 11468-11478.

Gemino, A., and Wand, Y. 2004. "A Framework for Empirical Evaluation of Conceptual Modeling Techniques," *Requirements Engineering* (9:4), pp. 248-260.

Gemino, A., and Wand, Y. 2005. "Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties," *Data & Knowledge Engineering* (55:3), pp. 301-326.

German Asphalt Pavement Association. 2011. *Guidelines to ensure the usable lifetime of hot mix asphalt pavements*, Bonn: DAV.

Johnson-Laird, P. N., and Byrne, R. M. J. 1991. *Deduction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Kaiya, H., and Saeki, M. 2006. "Using Domain Ontology as Domain Knowledge For Requirements Elicitation," in *Proceedings of the 14th IEEE International Conference on Requirements Engineering (RE'06)*, New York: IEEE Press, pp. 189-198.

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. 2007. "Ontology Visualization Methods – A Survey," *ACM Computing Surveys* (39:4), Article 10.

Khatri, V., Vessey, I., Ramesh, V., Clay, P., and Park, S.-J. 2006. “Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge,” *Information Systems Research* (17:1), pp. 81-99.
Lanzenberger, M., Sampson, J., and Rester, M. 2010. “Ontology Visualization: Tools and Techniques for Visual Representation of Semi-Structured Meta-Data,” *Journal of Universal Computer Science* (16:7), pp. 1036-1054.
Larkin, J. H., and Simon, H. A. 1987. “Why a Diagram is (Sometimes) Worth Ten Thousand Words,” *Cognitive Science* (11:1), pp. 65-100.
Lindland, O. I., Sindre, G., and Sølvberg, A. 1994. “Understanding Quality in Conceptual Modeling,” *IEEE Software* (11:2), pp. 42-49.
Mendling J., Reijers H. A., and Recker J. 2010. “Activity Labeling in Process Modeling: Empirical Insights and Recommendations,” *Information Systems* (35:4), pp. 467-482.
Motta, E., Peroni, S., Gómez-Pérez, J. M., d’Aquin, M., and Li, N. 2012. “Visualizing and Navigating Ontologies with KC-Viz,” in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi (eds.), Berlin: Springer, pp. 343-362.
Parsons, J. 2011. “An Experimental Study of the Effects of Representing Property Precedence on the Comprehension of Conceptual Schemas,” *Journal of the Association for Information Systems* (12:6), pp. 401-422.
Parsons, J., and Cole, L. 2005. “What Do the Pictures Mean? Guidelines for Experimental Evaluation of Representation Fidelity in Diagrammatical Conceptual Modeling Techniques,” *Data & Knowledge Engineering* (55:3), pp. 327-342.
Parsons, J., and Wand, Y. 2008. “Using Cognitive Principles to Guide Classification in Information Systems Modeling,” *MIS Quarterly* (32:4), pp. 839-868.
Poels, G., Maes, A., Gailly, F., and Paemeleire, R. 2011. “The Pragmatic Quality of Resources-Events-Agents Diagrams: An Experimental Evaluation,” *Information Systems Journal* (21:1), pp. 63-89.
Protégé. 2016. Protégé 4 user’s guide (online at http://protegewiki.stanford.edu/wiki/Pr4_UG; retrieved September 8, 2016).
Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., and Wroe, C. 2004. “OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns,” in *Proc. of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, E. Motta, N. R. Shadbolt, A. Stutt, and N. Gibbins (eds.), Berlin: Springer, pp. 63-81.
Rips, L. J. 1983. “Cognitive Processes in Propositional Reasoning,” *Psychological Review* (90:1), pp. 38-71.
Robillard, P. N. 1999. “The Role of Knowledge in Software Development,” *Communications of the ACM* (42:1), pp. 87-92.
Shanks, G., Tansley, E., Nuredini, J., Tobin, D., and Weber, R. 2008. “Representing Part-Whole Relations in Conceptual Modeling: An Empirical Evaluation,” *MIS Quarterly* (32:3), pp. 553-573.
Sharman, R., Kishore, R., and Ramesh, R. 2004. “Computational Ontologies and Information Systems II: Formal Specification,” *Communications of the Association for Information Systems* (14), Article 9.
Smith, E., and Medin, D. 1981. *Categories and Concepts*, Cambridge, MA: Harvard University Press.
ter Hofstede, A. H., and van der Weide, T. P. 1993. “Expressiveness in Conceptual Data Modelling,” *Data & Knowledge Engineering* (10:1), pp. 65-100.
Tiwana, A. 2004. “An Empirical Study of the Effect of Knowledge Integration on Software Development Performance,” *Information and Software Technology* (46:13), pp. 899-906.
Vessey, I. 1991. “Cognitive Fit: A Theory-Based Analysis of the Graphs versus Tables Literature,” *Decision Sciences* (22:2), pp. 219-240.
W3C. 2004. *Web Ontology Language Reference* (online at http://www.w3.org/TR/owl-ref; retrieved Septembe 8, 2016).
Wand, Y., and Weber, R. 1993. “On the Ontological Expressiveness of Information Systems Analysis and Design Grammars,” *Information Systems Journal* (3:4), pp. 217-237.
Wand, Y., and Weber, R. 2002. “Information Systems and Conceptual Modeling – A Research Agenda,” *Information Systems Research* (13:4), pp. 363-376.
Warren, P., Mulholland, P., Collins, T., and Motta, E. 2014. “The Usability of Description Logics,” in *Proceedings of the 11th Extended Semantic Web Conference (ESWC 2014)*, V. Presutti, C. d’Amato, F. Gandon, M. d’Aquin, S. Staab, and A. Tordai (eds.), Berlin: Springer, pp. 550-564.