Automated Ontology Design

Keith Allen
Candidate for B.S. Degree
in Applied Mathematics and Computer Science

State University of New York at Oswego
College Honors Program
December, 2023

Abstract

As the field of Ontology grows, and the need for ontologies increases, better tools are needed to create ontologies. In this paper we cover a background on domain ontologies, Basic Formal Ontology, and the benefits of ontology. We then present one such system being developed to improve ontology creation, the Dialog Based Ontology Learner, the methodology behind the DBOL's creation, and the current work being done on the project. Finally, two rounds of user tests are presented, finding that users are able to classify terms to their corresponding BFO superclass 25%-35% of the time. These preliminary results show that training for the system, as well as the formatting of questions being asked, must continue to be improved and refined, as well as that the system is step in the right direction for automated ontology design.

Contents

List of Figures

1 Acknowledgments

I want to thank Dr. Roman Ilin and Dr. Shane Babcock for their work and guidance on the DBOL team. The experience of working as part of a small and dedicated team for the past year has been amazing and impactful. Their knowledge of Ontology has been the source of much of my knowledge on the subject.

I would like to thank all of the test users both from this summer at ATRC, and this semester when we ran the test again at SUNY Oswego. Your input has been invaluable.

Of course I would like to thank my family who has always supported me. Additionally thank you to my friends who have had to listen to me ramble about ontologies for the past year, especially for the more than a few practice runs of presentations.

I would like to thank Dr. Dan Schlegel for his guidance as both the primary advisor of this honors thesis, as well as throughout my time at SUNY Oswego.

Thank you to Dr. Elizabeth Wilcox for her guidance as the secondary advisor for the honors thesis, as well as my capstone advisor.

I know even if I wrote a thesis length of thanks, there would still people I am forgetting to thank here, so thank you.

2 Introduction and Background

2.1 What is an Ontology?

The story of Ontology is the story of sorting, cataloging, and classifying the world around us. It is a subfield of Metaphysics, which in turn is a branch of Philosophy. In its most general sense, the field of Ontology is concerned with creating a taxonomy of entities and the relations between the entities. The repositories are then in turn referred to individually as an ontology. These classifications and relations are universal and aim to represent reality in a consistent repository [5]. To build ontologies, *terms* are assigned a **superclass**[1] within an ontology. These superclasses are also referred to as the parent of the term. This is a monumental task, and in the process of tackling it, many types of ontologies have been developed. Two are of concern to this project.

The first type are top level ontologies (TLO). These ontologies classify terms in high level, general categories. For example, there are the Basic Formal Ontology (BFO) terms **object** and **process** [5], an **object** being the paper this printed on, or a **process** being someone reading this paper. Because these classifications are so general, and there are so few native BFO terms, that at first glance terms assigned to a superclass may not appear to have much in common get grouped together. For example, in the Food Ontology (FoodOn), *Liquid* and the *consumer-ready food packaging* both are subclasses of the BFO **material entity** superclass. To get into the specifics, domain ontologies are needed.

Domain ontologies are more granular, and as their name implies, are built to classify terms in a specific domain. Some examples of domain ontologies include the Ontology of Electronics (OOE) and the Epilepsy Ontology (EPIO). Instead of large overarching classifications, domain classes can become incredibly specific, such as the *food condiment product* class within the (FoodOn) [7]. Recalling the example, *Liquid* and the

[1] Throughout this paper, terms assigned to ontologies are presented using *italics*, and ontological classes using **bold text**.

consumer-ready food packaging from the FoodOn, within this domain ontology more context is added so these terms are not just both subclasses of **material entity**, but informative classifications that are direct children of other domain ontology terms that eventually meet at **material entity**. When using best practices to create ontologies, all domain terms are descendants of TLO terms, and together they can create highly consistent and detailed ontologies.

2.2 Importance of Ontologies

Now that the goal of Ontology, and the types of ontologies, have been outlined, an important question arises. Why put in the time and effort to study and create ontologies? The answer to this question lies within data analysis and the ever-increasing utilization of large data sets. The use of ontologies as a whole allows for consistent semantic labeling of large data sets. By labeling the elements of these massive data sets, they become easier to compute over, and better insights can be drawn from them [5]. Not only can more be learned from consistently labeled data, but this practice also allows data to be reused and shared both within, and outside of, the original organization or domain. These benefits drawn from using ontologies have been noticed and in response, ontologies are being developed both by companies and the government in an attempt to help standardize ever-expanding information systems [2].

However, the pairing of top level and domain ontologies was not always the norm. The modern field of Applied Ontology traces its origins within Bioinformatics [5]. Decades ago experts in different domains of Bioinformatics started building ontologies, but when trying to share their ontologies they ran into the issue of siloing. As good as any of the ontologies were individually, there was nothing connecting them. Instead each ontology existed on its own, separated into silos by this lack of centralized agreement. On their own, each ontology

[2] Along with increased interoperability there is the beginning of a Ontological Reasoning. The eventual goal is that BFO compliant ontologies will be capable of making inferences. Currently, due to the underlying description logic used to query ontologies, some statements that the team believes are crucial for practical ontology applications cannot be formed. This work is in its rudimentary stages.

was an outstanding resource to its organizations, but they lost out on the benefits of sharing information and ontology terms amongst themselves. In an effort to combat this, TLOs were introduced. While there are multiple common TLOs, this project focuses on BFO.

BFO is the first TLO recognized by the International Organization for Standardization (ISO). It was recognized as ISO/IEC 21838-2:2021 in November of 2021 [3], but even before becoming an ISO standard BFO had been widely adopted throughout the world, heavily so within Bioinformatics. This is, of course, because of its close work and development alongside Bioinformatics since the early 2000's. According to the official BFO website, BFO is currently utilized by over 400 ontologies by over 100 organizations [7]. Its widespread implementation makes it a great candidate to explore the possibilities of assisting users with automation. By using a TLO that is already standardized the products of this research can be utilized by an established community.

2.3 Project Motivation

It has been established that ontologies can offer great improvements when computing over large data sets and the benefits are increased by tying together domain ontologies via TLOs. So why are there not domain ontologies for every domain and all of their relevant terms already? The reality of creating domain ontologies is that it is an arduous process that takes a great length of time, along with close collaboration between domain experts and ontologists. The issue is that ontologists do not know enough about specific domains and domain experts do not know enough about ontology, making any separate efforts from either unfruitful. The most effective solution is to have domain experts work closely with ontologists so that the structure of the ontology is correct, and integrated with other ontologies, as well as the terms are correctly assigned. However, any process that takes this level of teamwork becomes slow and costly. In addition to the amount of labor required, the process is also prone to human error and if not created correctly, an ontology might have circular relations between terms. Sets of two terms that form these circular relations can be easy to spot with the human eye, but cycles containing three terms or more can be hard to spot [9].

The solution to these problems lies in creating better tools that assist users in correctly building ontologies, in addition to the current tools used to build and display ontologies. This is exactly what this project is attempting to do. This team has worked to create a Dialog Based Ontology Learner, a computer system that can be used by users without ontology experience to create BFO compliant domain ontologies. The system asks questions, whose answers guides further questions, to help the user match terms to superclasses. During my time with the team the system was converted from asking “Yes”/“No” questions to asking questions with natural language answers.

During this development two rounds of user testing were administrated in an attempt to measure how average users were classifying terms. Along with measured data test users were able to provide feedback on the user experience of the system. Both this quantitative and qualitative feedback was used, or is going to be used, to make improvements to the system, until it is capable of assisting a user as it should.

2.4 BFO

While BFO has been introduced, it is important to have a better understanding of what it is, especially the phrase “BFO compliant”. BFO was developed by a team led by Barry Smith and is meant to help create connective fibers between domain ontologies [5]. When a user assigns a term a BFO classification, it is then represented as a subclass of the chosen BFO superclass. These terms can then be considered to be leaves of BFO, but within domain ontologies these terms are often nested under one another. From the previous example *food condiment product* and *jelly condiment* both belong to the BFO **material entity**, but *jelly condiment* is nested under *food condiment product* within the FoodOn [2].

With that out of the way, what does it mean to be “BFO compliant”? An ontology is

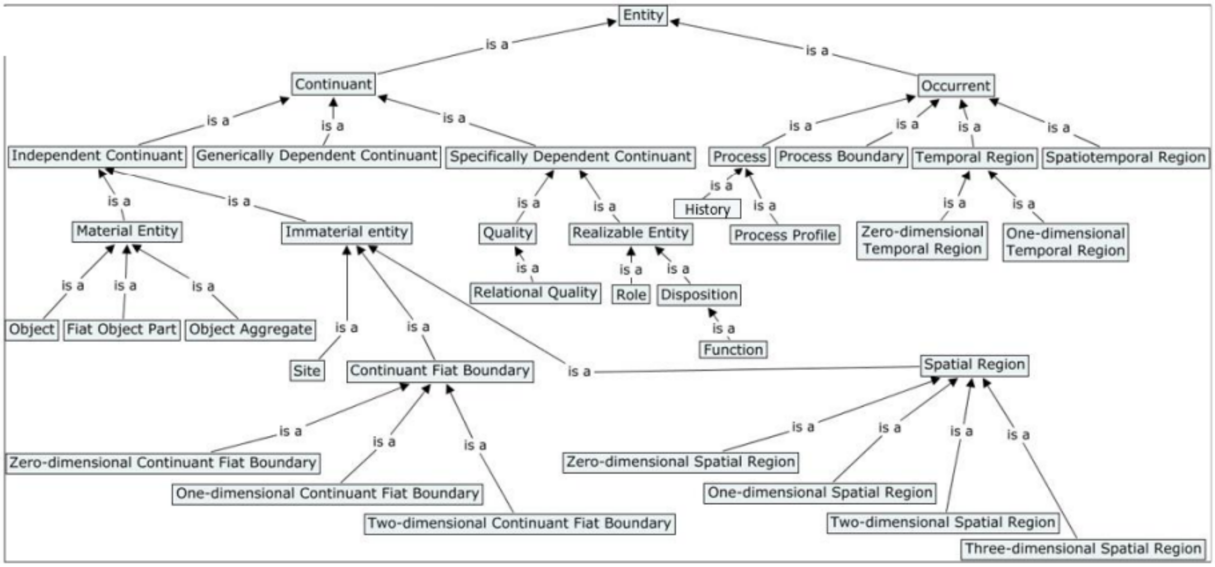

Entity
is a
Continuant
Occurrent
Independent Continuant
Generically Dependent Continuant
Specifically Dependent Continuant
Process
Process Boundary
Temporal Region
Spatiotemporal Region
History
Process Profile
Zero-dimensional Temporal Region
One-dimensional Temporal Region
Material Entity
Immaterial entity
Quality
Realizable Entity
Relational Quality
Role
Disposition
Function
Object
Fiat Object Part
Object Aggregate
Site
Continuant Fiat Boundary
Spatial Region
Zero-dimensional Continuant Fiat Boundary
One-dimensional Continuant Fiat Boundary
Two-dimensional Continuant Fiat Boundary
Zero-dimensional Spatial Region
One-dimensional Spatial Region
Two-dimensional Spatial Region
Three-dimensional Spatial Region

Figure 1: BFO 2.0 Hierarchy from BFO 2.0 Specification and User Manual [8]

BFO compliant when all of its terms belong to a BFO superclass. By building BFO compliant domain ontologies, it helps to bring standardization to the ontology. By requiring all terms to belong to a BFO superclass it ensures between ontologies at least at some level these terms share features. Even at this seemingly high level, classifying these terms still tells the user a lot of information. Everything that is a BFO **object** is known to be made of matter and to have a clear boundary to it. Likewise, if a term is an **information content entity** then it might not be information, and likely instead conveys, describes, or represents an idea or some information. This information is still useful to humans and computers, and gives different domain ontologies some level ground to all agree on. The full BFO 2.0 hierarchy can be found in figure 1.

3 The DBOL

3.1 Intro to the DBOL

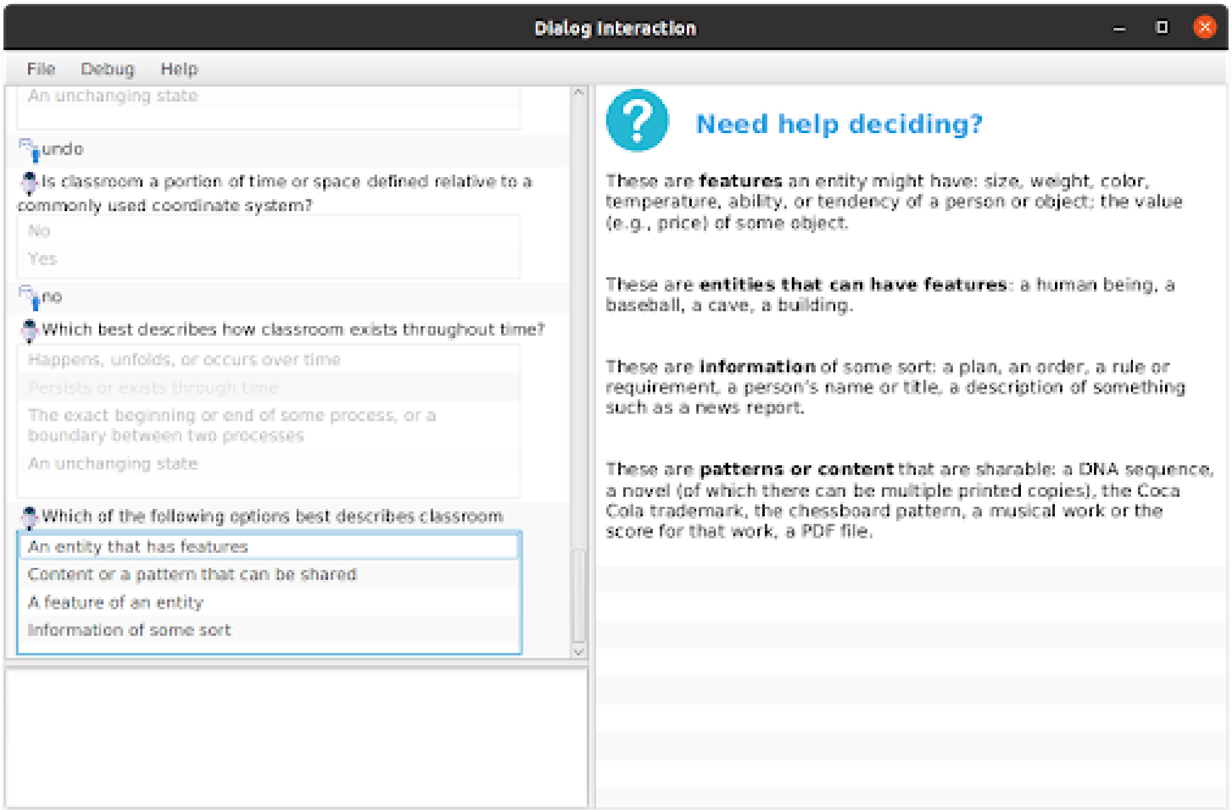
In an attempt to solve the described problems a team has been working for the last two years developing the Dialog Based Ontology Learner (DBOL). The DBOL is a standalone software system that utilizes a Java back end, JavaFX front end, and has an optional Amazon Web Services connection for log storage. The system uses current trends in Artificial Intelligence, such as dialog systems and analogical reasoning, to assist a domain expert in classifying terms to create an ontology. As the user continues to enter and assign terms the ontology is stored in a CSNePS knowledge base.

The DBOL team consists of four members. Dr. Roman Ilin, an Air Force Research Lab (AFRL) computer scientist with a background in sensor technology and an interest in the role ontologies can play in intelligence and improving artificial intelligence reasoning. Dr. Shane Babcock, an ontologist who studied under BFO co-creator Barry Allen, ensures that the DBOL is being correctly programmed to make BFO compliant ontologies that can be integrated with other ontologies. Dr. Dan Schlegel, a logical reasoning expert who developed the CSNePs system. He wrote and maintains the DBOL code base and guides its continued development. I joined the team roughly a year and a half into development, and at that point much of the system was created. By the time I joined, the DBOL was able to classify terms to their BFO superclasses, and the next step was to adjust that process to bring the project closer to its goal of having a conversational dialogue with the system. As part of this team, I was responsible for implementing updates to the system based on weekly team meetings. While these weekly changes were mostly implementing changes to the questions being asked, as well as expanding user interface options to new questions, it created a cycle of tweaking and then testing within the team. I also found, documented, and attempted to fix bugs, as well as updated outdated JUnit tests.

Throughout the summer I also started documenting the system, creating an updated user manual for CSNePS, as well as beginning documentation for a user manual, developer manual, and script collection for the DBOL. Finally, I also helped develop, get approval for, administer, and lastly analyze user testing of the DBOL at the end of the summer and again in fall. The two rounds of user testing are informing next steps for the DBOL team, and give chances to improve the system.

CSNePS, or Concurrent SNePS, is a knowledge representation and reasoning system (KRR) developed by DBOL team member Dr. Schlegel as part of his PhD thesis. The system is a Clojure port, and expansion, of the SNePS 3, a KRR system developed by the SNePS research group [1]. By implementing in Clojure CSNePS the system is able to utilize Java concurrency, as well as easily integrate with Java systems. However, ontologies are typical stored in Web Ontology Language (OWL). The OWL language was developed by the OWL Working Group in the early 2000s, and was created to utilize the structure and versatility of XML in a semantic domain [4]. While it has been adopted as the standard for storing ontologies and the semantic web, OWL can only captures binary relations. Comparatively, CSNePS is able to capture relations between n terms where $2 \leq n$. Therefore, it is easy to do logical computations if the ontology can be stored in CSNePS. CSNePS also uses backwards reasoning to find new, logically valid, relations when other new relations are added to the knowledge base. This is computationally feasible due to Clojure's use of Java concurrency, and allows for the knowledge base to grow, without solely relying on user input.

The DBOL itself has 4 distinct sections, all laid out to provide a straightforward user experience, all of which can be seen in figure 2. At top of the DBOL sits the Menu Bar which can be used to help navigate the session. From here a user can reload a previously saved session with the DBOL allowing a user to pick up right where they left off. Additionally a user can open the CSNePS visualizer to see in real time how the knowledge base is growing as they add terms. In the upper left sits the Chat Box. The Chat Box is



An unchanging state

undo

Is classroom a portion of time or space defined relative to a commonly used coordinate system?

No

Yes

no

Which best describes how classroom exists throughout time?

Happens, unfolds, or occurs over time

Persists or exists through time

The exact beginning or end of some process, or a boundary between two processes

An unchanging state

Which of the following options best describes classroom

An entity that has features

Content or a pattern that can be shared

A feature of an entity

Information of some sort

Need help deciding?

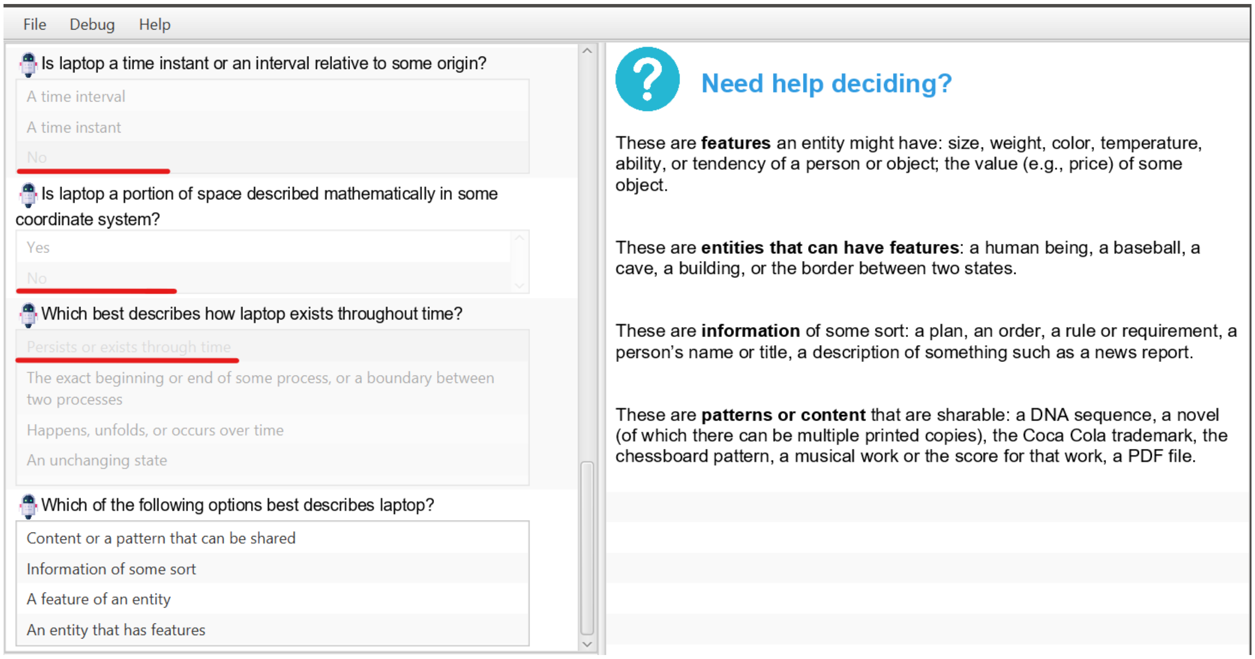
These are **features** an entity might have: size, weight, color, temperature, ability, or tendency of a person or object; the value (e.g., price) of some object.

These are **entities that can have features**: a human being, a baseball, a cave, a building.

These are **information** of some sort: a plan, an order, a rule or requirement, a person's name or title, a description of something such as a news report.

These are **patterns or content** that are sharable: a DNA sequence, a novel (of which there can be multiple printed copies), the Coca Cola trademark, the chessboard pattern, a musical work or the score for that work, a PDF file.

Figure 2: DBOL layout while classifying *classroom*.



Is laptop a time instant or an interval relative to some origin?

A time interval

A time instant

No

Is laptop a portion of space described mathematically in some coordinate system?

Yes

No

Which best describes how laptop exists throughout time?

Persists or exists through time

The exact beginning or end of some process, or a boundary between two processes

Happens, unfolds, or occurs over time

An unchanging state

Which of the following options best describes laptop?

Content or a pattern that can be shared

Information of some sort

A feature of an entity

An entity that has features

Need help deciding?

These are **features** an entity might have: size, weight, color, temperature, ability, or tendency of a person or object; the value (e.g., price) of some object.

These are **entities that can have features**: a human being, a baseball, a cave, a building, or the border between two states.

These are **information** of some sort: a plan, an order, a rule or requirement, a person's name or title, a description of something such as a news report.

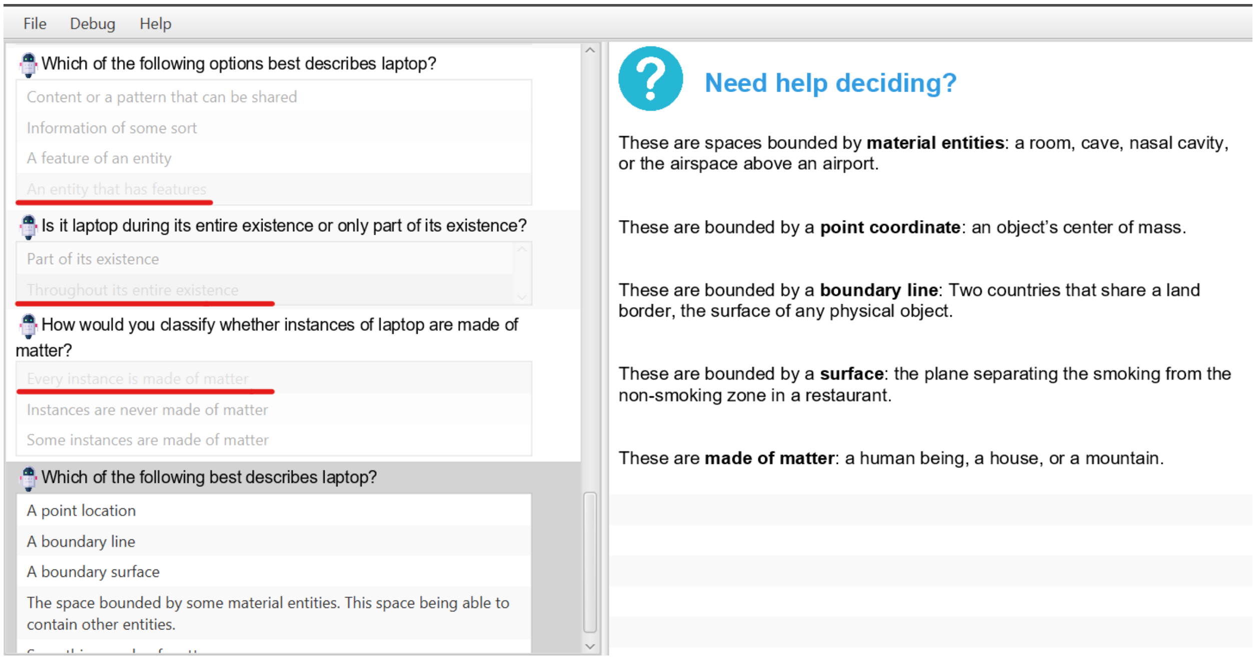
These are **patterns or content** that are sharable: a DNA sequence, a novel (of which there can be multiple printed copies), the Coca Cola trademark, the chessboard pattern, a musical work or the score for that work, a PDF file.

Figure 3: First three questions to classify *laptop*.

where the DBOL's questions, answer options, and the users answers will appear. Many questions allow for selection from a list, which will be found in the Chat Box. After the first few messages a scroll bar will appear that can be used to look through the past messages of the users session. The lower left is known is the Text Box, which allows for free form textual input from the user. This is used when a question does not have a list selection answer, when the user is entering a new term, and when a user wants to roll back a selection. Finally, the right portion of the DBOL is the Tip Box. This holds examples, suggestions, and other useful information that the user might need while using the DBOL. For a more detailed introduction to the DBOL and its functionality see the DBOL User Manual for Testing as an appendix 5.3.

Let's work through an example of the system being used to classify the term *laptop*. Throughout figures 3, 4, and 5 the user answers nine questions to classify the term. The answers were selected from the lists below questions. Since the selected answer can be hard to read they have all been underlined in red to assist the reader.

In the first few questions the DBOL is attempting to determine whether *laptop* is an



Which of the following options best describes laptop?

- Content or a pattern that can be shared
- Information of some sort
- A feature of an entity
- An entity that has features

Is it laptop during its entire existence or only part of its existence?

- Part of its existence
- Throughout its entire existence

How would you classify whether instances of laptop are made of matter?

- Every instance is made of matter
- Instances are never made of matter
- Some instances are made of matter

Which of the following best describes laptop?

- A point location
- A boundary line
- A boundary surface
- The space bounded by some material entities. This space being able to contain other entities.

Need help deciding?

These are spaces bounded by **material entities**: a room, cave, nasal cavity, or the airspace above an airport.

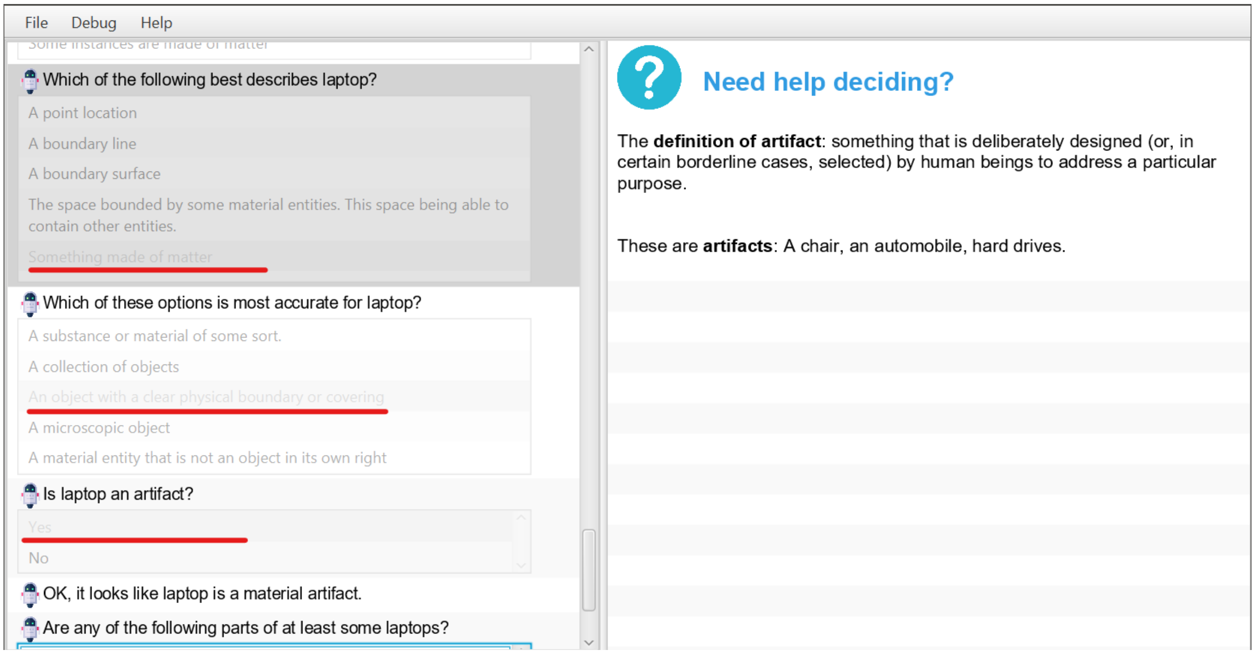
These are bounded by a **point coordinate**: an object's center of mass.

These are bounded by a **boundary line**: Two countries that share a land border, the surface of any physical object.

These are bounded by a **surface**: the plane separating the smoking from the non-smoking zone in a restaurant.

These are **made of matter**: a human being, a house, or a mountain.

Figure 4: Second set of three questions to classify *laptop*.



- Some instances are made of matter

Which of the following best describes laptop?

- A point location
- A boundary line
- A boundary surface
- The space bounded by some material entities. This space being able to contain other entities.
- Something made of matter

Which of these options is most accurate for laptop?

- A substance or material of some sort.
- A collection of objects
- An object with a clear physical boundary or covering
- A microscopic object
- A material entity that is not an object in its own right

Is laptop an artifact?

- Yes
- No

OK, it looks like laptop is a material artifact.

Are any of the following parts of at least some laptops?

Need help deciding?

The **definition of artifact**: something that is deliberately designed (or, in certain borderline cases, selected) by human beings to address a particular purpose.

These are **artifacts**: A chair, an automobile, hard drives.

Figure 5: Final three questions to classify *laptop*.

occurrent, or something that is related to time itself, like a day. Then the system asks how *laptop* "exists through time". This question is meant to assess whether the term is something that exists through time or something that is happening throughout time. Since *laptop* exists throughout time, "persists or exist through time" is chosen.

Now the DBOL is determining if the term is a entity or if instead the term describes an entity in some way. Once the DBOL knows that *laptop* is an entity it needs to know if the term is always that thing, or if there are times when it is not. For example, people are always people, but not always their profession. However, a laptop itself is always a laptop, so we can continue. The last answered question in figure 4 attempts to figure out if the term is always made of matter. Since laptops are always made of matter, this is straightforward.

The last few questions attempt to do some final narrowing. At this point the DBOL knows that *laptop* is an entity that is made of matter, a good start, but still there are better specifics to find. The seventh question wants to make sure the user means that the term is an object and not a bounding part of the object. Finally, the DBOL needs to know if laptop is an object by itself or if it is part of a larger system, or if it is an object at all. Since laptops are clearly separate from a larger system, as opposed to say *parts of an engine*, it has a clear physical boundary. The final question does technically stray from BFO asking if *laptop* is an artifact. The tip can still be seen on the right where artifact was defined. Since *laptop* is something made with a specific purpose it is, in fact, an artifact.

Finally, *laptop* has been assigned a superclass, and the DBOL can then be seen asking follow up questions. These questions are unique to the standard mode. In the evaluation mode, these additional questions are omitted and the system stops asking once the term has been assigned a superclass.

3.2 Work on the DBOL

As detailed in previous sections, the DBOL is being developed to serve the role of building full domain ontologies. These ontologies need to be BFO compliant, meaning that every domain term needs to be assigned to a BFO superclass before the DBOL can start to sort the terms into their corresponding domain superclasses. To tackle this task, the DBOL team has written a set of questions known as the beginner rules. For the first year of the project the beginner rules were a set of "Yes"/"No" questions that, while functional, were clunky and detached from how humans tend to converse with one another. This distanced the DBOL from its goal of usability and did not align with the goal of using a dialog system, meaning change was needed.

The first major adjustment to the beginner rules saw the change from "Yes"/"No" questions to a set of 20 questions, capable of assigning terms to 31 superclasses with a maximum depth of 9 questions for any given term[3]. These new questions used natural language answers which aided in giving DBOL sessions a more conversational feel as a user. More importantly, by creating a question tree that mimicked the structure of the BFO hierarchy, the system was able to navigate users to the correct classification in fewer questions, since one question was able to rule out large sections of possible superclasses.

To account for the new natural language answers the user interface (UI) needed to be changed. Selection from a list was added so that users could answer questions more easily, and this also removed the worry of mistyping answers. Along with the UI, the goal engine and scripts were expanded and extended to account of the new questions, now known as abstract questions, in contrast to the former binary questions.

Overall, the change from binary questions to the natural language questions moves the DBOL closer to its eventual goal of importing and creating full fledged domain ontologies.

[3] Some of the classifications in the beginner rules are more specific than the BFO superclasses. Superclasses such as **material artifact**, and **directive information content entity** are from the Common Core Ontology (CCO) and their use is widespread enough that incorporating them hopes to save the user time in a later versions of the DBOL.

3.3 Similarity Metrics

As the DBOL geared up for the first round of user testing, a question was raised: "What does it mean to assign a term correctly"?[4] The answer is straightforward. A term is assigned correctly when it meets the criteria of its specific superclass. For example if a term that is always made of matter is assigned as an **immaterial entity**, it would not fit the criteria of not being made of matter. However, this led to a more important question of "What does it mean to get term assignment wrong"? There is the simple "correct" or "incorrect", but there is more meaning to be found than just "yes" or "no". On the other hand this is not a call that can just be made subjectively where some term assignments "feel more correct". This project is about providing structure, subjective decision-making would do the opposite. Instead there needs to be a measurement that can capture how correct or incorrect a term assignment is within an Ontology. In the Fall I began working with Dr. Wilcox to attempt to measure the incorrectness of assignments. A possible solution lives within similarity metrics.

Due to ontologies being directed relations with a subsumption hierarchy, they can be abstracted into directed acyclic graphs, or DAGs[5]. While all domain ontologies have not been proved to be DAGs, the BFO ontology, and any terms added as leaves, is a DAG. Since all user testing being done assigned terms directly off of BFO, then the collection of all user results is also a DAG. Now that the results can be abstracted into a mathematical structure, it is easier to compute over them.

A similarity metric is needed that considered the directed nature of DAGs, and can include the importance of the direct parent child relation. For this, Katz Similarity (KS) is used. It fits the above criteria, and has already been implemented in C for use in studying knowledge hierarchy evolution [6]. However, KS is computationally expensive, since every

[4] This section has been adapted from my capstone paper in which I explored ways to measure two domain ontologies with the same terms, but different "is_a" relations. The paper, along with the code can be found at https://github.com/KeithTAllen/Katz-Similarity.

[5] This is when considering the BFO "is_a" relation, the relation that relates a term to a superclass. For example, *laptop* "is_a" **object**.

node in a DAG needs to be compared to every other node, these individual KSs are then used to compute the Katz Graph Similarity, a measurement of similarity between two graphs. The computational cost however can be reduced by calculating the KS of nodes recursively while performing a breadth-first graph traversal. This method has been implemented in Java to match the rest of the DBOL project, but over large ontologies it is still slow, and loses precision, resulting in faulty output. The true solution is to approximate the Katz Graph Similarity. This method looks at small subgraphs of the larger DAG, saving both time and precision. This approximation of the KGS is set to be implemented next to increase precision and speed as the DBOL prepares to start accepting the import of ontologies.

4 User Testing

4.1 User Testing Setup and Goals

After refining the new beginner rules, two user tests were conducted with changes made in between based on the results of the first user test. The user tests primary purpose was to test how effective the beginner rules were at helping a novice user, with little to no Ontology experience, correctly classify terms to their appropriate BFO superclass. However, there was a secondary purpose: these tests should gauge the DBOL's usability along with the documentation meant to teach a user how to use the DBOL. With these two goals in mind a user testing plan was developed, approved, and administered.

An evaluation mode was developed for the DBOL specifically for user testing. Instead of the user flow of entering terms, classifying the terms into the BFO hierarchy, and then answering followup questions to begin creating a domain ontology, the evaluation mode took an input csv file containing terms and their respective definitions. Additionally once the terms were assigned to their BFO superclass, the program moved on to the next term, skipping over any follow up questions that did not concern BFO. The evaluation mode also recorded the user's classification and created an output csv containing the terms, their definitions, and the user assigned superclass. This special evaluation mode also holds possibilities for creating ontologies in which a user could use input csv files with set terms and definitions to help create their ontology.

For the first round of user testing each volunteer was asked to perform three tasks. First, the users were asked to read through an informed consent form to learn more about the project, what they were being asked to do, and any risks related to their participation. Secondly, the user read through a shortened version of the DBOL User Manual. This manual was abbreviated from the full User Manual by omitting information and functionality that the test users would not encounter or use. Finally, when the users were ready they used the DBOL to classify five terms.

Questions were welcomed throughout the testing, but questions about the term's or the questions being asked were not answered. Since the primary goal was to test how effective the beginner rules were, not how easy it was the use the DBOL or how effective the training was, by not answering questions on how to use the system the results might be impacted by usability, and not the beginner rules themselves. The secondary goal of testing the usability of the system was instead measured by an exit survey.

The terms that the users were classifying were chosen from a list of 1000 common English nouns, whose definitions were then pulled from WordNet. Each user classified five terms, and to test both a wide variety of words, as well as how different users classify the same terms, an overlapping pattern of test words was developed. For any given user, the first term they classified was the last term of the previous tester's words, and their final term was the first term that the next user classified. However, since the classification of a term depends entirely on its definition, users needed to be able to change the definition of a term. For example consider the term *chip* with the definition "to remove small pieces of" in comparison to the definition "a crunchy snack often made of potatoes". The first **chip** is a BFO **process**, while the second is an **object**. When each word was introduced, it was accompanied by a definition and possibly some examples from the WordNet entry of the term. Users were prompted to change the definition if it did not match the term or if they wanted to classify a different version of the term. Whichever definition was used to classify the term, either given or provided, was included in the output file. This way when agreeing upon a classification, the DBOL team and the user were classifying the same term.

In both instances of user testing not all of the terms were usable. A known bug in the system would cause a user's term to be overwritten by the previous term, meaning that the user's classification of some terms was omitted from the results[6]. Terms were also omitted if the definition was too vague, or if multiple conflicting definitions were provided. The last

[6] This bug was caused by users attempting to undo over term assignment, a known problem that was outlined in the user manual, but was still routinely performed during testing.

reason a term was removed is if it was a reserved BFO word, or did not have a classification within BFO. One such term was *square*. BFO is not well-equipped to handle mathematical terms like *square*, but work is being done to develop the Math Ontology which will help alleviate this issue. In both cases of testing few words were deemed unusable, meaning that there were still plenty of terms and classifications to analyze.

After each round of user testing, the project members all met and, before revealing what the users classified the terms as, all agreed upon a classification for each term considering its definition. Only after a classification was assigned were the user answers revealed, and their subsequent comparison against the agreed-upon classification was then possible. In rare cases a term was determined to fit into two categories depending on what exactly the user was thinking, all of which relied on complicated ontological patterns between superclasses in which when you have a term that is classified as **x** then there is a similar term that belongs to class **y**. For example, *temporal interval* and **measurement information content entity** are often found together. There is the term *day* which refers to the time within a day as contrasted by the term *day* that refers to the length of a day in hours[7]. In this case either the term was given two agreed-upon classifications, but this was rare enough not to impact the results. This is a different case than providing a term with two definitions, since at its core the term shares a definition, there are just multiple aspects of this term that need to be considered.

4.2 Summer Testing Results

The first run of user tests were performed in Dayton, Ohio, and the volunteers were all summer interns or personnel from the Autonomy Technology Research Center. Fourteen volunteers classified 68 terms in total, and out of those terms 25 of the user classifications matched the agreed classification for a success rate of 37 percent. For such a small sample size and a preliminary test these results showed promise, but two cases stuck out.

[7] This is an area of improvement planned for the DBOL. The hope is that the system can detect when a user classifies a term that likely has a related term and will ask about this newly detected term.

First, users incorrectly classified many terms as **occurrent** or its subclasses. This meant that the user had answered the first question incorrectly and been led down a path they could not get off of in time before assigning their term. Second, users classified many terms as **stasis** despite that being one of the more niche categories. This led the team to reevaluate the question that classifies a term as a stasis.

Oddly enough, the team noticed that an unusual number of the terms were classified as **process**. Nineteen of the terms were assigned an agreed classification of **process**, and more importantly users assigned those terms correctly 12 times. This meant that processes were almost twice as likely to be assigned correctly than any given term.

The first run of user testing showed that this proof of concept was feasible, as well as affirmed that this is a subject that people were interested in, with a system that users are capable of utilizing.

4.3 Changes Following Summer Testing

With the newly gained insights from the preliminary test, there were some changes that needed to be made. In between the two sets of user testing the following was changed within the DBOL's evaluation mode.

In an attempt to help correct those who were accidentally classifying terms as **stasis**, the corresponding question was adjusted, its tip box was expanded, and additional questions were added to check that the user understood what a stasis was and that they then still wanted to assign the term as a stasis. This "guardrail", as it was called, within the beginner rules was common in the original binary rules, but was omitted from the natural language beginner rules. With the thought being, that if the user is able to use more complex descriptive answers, there would be no need to check.

After adding the stasis check, a similar "one size fits all" guardrail was created for every classification. Since the evaluation mode was pulled together from the standard DBOL mode adjustments had to be made to the goal engine to ignore its standard behavior in favor of the highly structured, now required behavior for the evaluation mode. This was most evident when using the undo functionality, since once a user assigned a superclass to a term, they could not undo it, and with no warning that an answer would cause a term to be classified, users could find that they had classified their term as a **stasis** or **temporal interval** in two questions before they could comprehend the term or what they had done. To prevent this, a general question was asked before a term was classified. The user was simply asked, “Okay. It looks like *term* is a **superclass**. Do you want to assign this term as a **superclass**?”. Unlike the stasis question, this “one size fits all” did not help to further define the superclass, instead it just made sure that users knew that they were about to assign a term, and thus would not be able to change their mind once they completed this action.

The opening question was also adjusted. Instead of asking about how whether a term was a “portion of time or space defined relative to a commonly used coordinate system?”, the question was reworked and combined with another to determine if the term was a **temporal interval** or **temporal instant**. This was implemented to still be able to quickly eliminate large sections of the answer space, while giving extra attention to the rarer superclasses, hoping that users would learn more about these superclasses and be less likely to blindly assign the terms as them.

The tip boxes to most questions were also expanded to improve both quantity and quality of the tips, and in another attempt to help users understand the difference between spatial regions, and entities that contain spatial regions a section about spatial regions was added to the end of the user manual for volunteers to read. This addition did not drastically extend the length of the user manual. The training required to user the DBOL is unlikely to drop to none, and so expecting that users can read, comprehend, and then apply basic ontology knowledge as part of a tutorial is not unreasonable.

4.4 Fall Testing Results

With all of these changes, and approval from the Oswego Human Subjects Committee, a second round of user testing was ready to begin. The second round of testing was held at Oswego and posters were hung around campus to collect volunteers, and thus the backgrounds of the volunteers was more varied. The test was administered in the same format of informed consent, user manual, followed by using the system, but now was finished with a short exit survey in hopes of collecting some formal user experience information. This section allowed users an anonymous space to mull over their time spent testing the system and provide feedback on what went well, and what they might have struggled with.

In total the second round of testing saw 32 volunteers who classified 150 terms, and of the 150 terms, 37 of the user classifications matched the agreed classification for a success rate of 25 percent. At a first glance this is not promising, since it shows that after all of these changes meant to improve the accuracy of classification, instead the opposite has occurred. However, there is still valuable information that can be gathered from this round of testing.

The two most popular incorrect classifications in the second round of testing were **temporal interval** and **temporal instant**. This is likely due to the opening question of the adjusted beginner rules. When adjusting the old opening question the following was instead introduced:

Is *term* a time instant or an interval relative to some origin?

Responses to which the users were presented with the choices of “A time instant”, “A time interval”, or “No”. Time after time, users became confused with this question and wouldn’t choose “No”, which then classifies the term, despite most of the possible superclass options only being accessible beyond this initial question. In fact, 13 users either never made it beyond the first question, or only saw another question one time. This problem was so prevalent that over half of the terms were classified as an instant or interval despite only nine of the terms had either as an agreed-upon classification. The wording of the question presents the usually correct answer as this odd, third, neither answer, likely impacting its selection rate.

However, when these users who only, or overwhelmingly[8], stuck to the first question, were set aside, the remaining users are able to offer insights into how effective the other questions were. Removing these users leaves 91 terms of which 33 were assigned correctly. Even within this set, 21 terms were still assigned as **temporal interval** or **temporal instant**, since users who chose either 3 or less times were still included. Even with temporal classifications being prevalent, this shows that the system is at least as effective as it was at the end of the summer, and if the issues surrounding temporal classification can be addressed, the correct classification could rise. Importantly, **stasis** was chosen less often, only being chosen for 2% of terms. It was, however, never chosen for terms that did belong in the category of **stasis**. While successful classification did not improve, users were at least less likely to classify terms as it incorrectly.

The main priority following the fall user testing is to resolve the issues surrounding the runaway number of temporal classifications. If the structure of the questions is to be preserved then the question needs to be reworked in a way that makes it clear that "Neither" is just as reasonable an answer as the other, easily defined answers. This can be done by rewording the question and adjusting the corresponding tips to help the user. Another possible course of action is to reconsider the structure of the questions and to take a similar path as the original beginner rules by starting with questions aimed at the more common superclasses such as **object** or **process**. This would allow for most terms to be assigned in fewer questions, but might present the opposite of the current problem. The results could show that rarer, but equally important, superclasses are being routinely being overridden by these common classifications.

[8] Chose **temporal interval** or **temporal instant** for four or more of their terms

Regardless of either path chosen, another improvement that could be made would be having users complete a short demo with the system. A two or three term practice test with the DBOL could help users who learn best with hands-on experiences, and offer opportunities to see the system in action, before being expected to classify terms correctly. As previously mentioned, expecting the DBOL to be simple enough to pick up that a user could approach it the same way they do a simple website is not reasonable. Creating ontologies is a specialized process, and those expected to be completing this task should expect that it will take time and training to use any associated tools. Balancing the barrier entry to the rigor required will need to be fine tuned by varying the amount and type of training new users receive, along with editing the functionality of the DBOL.

5 Future Work and Conclusion

Unfortunately, the second round of user testing did not live up to the team's expectations. While it did reveal more issues with the DBOL, more changes, and that another round of user testing will be needed. Eventually the team hopes to publish on the work done for this project. This will help boost automated ontology design, along with continue to support BFO as an emerging ISO standard.

The team will continue to make changes, expand the features, and run another set of user testing. Hopefully by restructuring or clarifying the questions users will be able to avoid the pitfalls of the second round of user testing. Eventually the goal is for the DBOL to become a system for full domain ontology creation, but there is still work to be done before this is realized. More work will continue to be done with similarity metrics, both improving the speed and widening their use.

Bibliography

[1] CSNePS Github. https://github.com/SNePS/CSNePS. Accessed: 2023-12-3.

[2] Food Ontology IRI. https://raw.githubusercontent.com/FoodOntology/foodon/master/foodon.owl. Accessed: 2023-12-19.

[3] ISO/IEC 21838-2:2021 Top-level ontologies (TLO) Part 2: Basic Formal Ontology (BFO). https://www.iso.org/standard/74572.html. Accessed: 2023-11-18.

[4] W3 Owl 2 Overview. https://www.w3.org/TR/owl2-overview/. Accessed: 2023-12-3.

[5] Robert Arp, Barry Smith, and Andrew D. Spear. *Building Ontologies with Basic Formal Ontology.* MIT Press, 2015.

[6] Guruprasad Nayak, Sourav Dutta, Deepak Ajwani, Patrick Nicholson, and Alessandra Sala. Automated assessment of knowledge hierarchy evolution: Comparing directed acyclic graphs. *Information Retrieval Journal*, 22(3–4):256–284, Aug 2019.

[7] Alan Ruttenberg. Basic Formal Ontology (BFO). https://basic-formal-ontology.org/. Accessed: 2023-11-18.

[8] Alan Ruttenberg. Basic Formal Ontology (BFO) Github Repository. https://github.com/BFO-ontology/BFO. Accessed: 2023-11-18.

[9] Selja Seppälä, Alan Ruttenberg, and Barry Allen. Guidelines for writing definitions in ontologies. *Ciência da Informação*, 46(1):73–88, 2017.

Appendix

5.1 Informed Consent Form

The next two pages are the informed consent form from the fall round of user testing. The summer informed consent form was omitted from the appendix since it was the same in terms of content, and only differed in format. The Oswego Human Subjects Committee heavily suggests a specific format for the informed consent form, and this the original form was rewritten to match this. The summer informed consent form was written based off of University of Dayton guidelines.

Date: 9/21/2023
Principle investigators: Keith Allen, Dr. Dan Schlegel
Study title: Automated Ontology Design

Informed Consent Document

My name is Keith and I am a student at SUNY Oswego conducting research on automated ontology design. The purpose of this study is to test the system that we have been developing (The DBOL). SUNY Oswego is familiar with this research and has given me permission to do this research. Also, the SUNY Oswego campus has a research oversight committee called the Human Subjects Committee, and they have also reviewed and approved this study. The purpose of this form is to inform you of details regarding this study so you can decide if you want to participate.

Your participation

Brief description of the project: If you choose to participate, you will complete the following: Complete this Informed Consent form, complete a short introduction to the DBOL, and then use the system to classify 5 random selected terms. Finally, we ask that you complete a short exit survey.

Your participation will take 15 minutes. The risks associated with participation are: There are no foreseen risks. The process involves reading, selecting, and typing. The words are pulled randomly from a list of 1000 common English words so none are expected to cause distress. You may withdraw from the study at any time. Doing so will not affect your relationship with the investigators or restrict you from any services or opportunities in the future.

Benefits

The benefits to you include: There are no direct benefits related to this experiment. The benefits of the research include: The creation of better ontology building tools which when can be used to create better domain ontologies.

Confidentiality

Personal information will not be collected or saved. Records from your DBOL session are stored securely and remotely, and contain no identifying data. This informed consent form will be securely stored by Dr. Schlegel. If the results from this research are presented at a conference or published in a journal, your information and individual responses will not be shared.

Questions?

If you have any questions about this study, please contact me at kallen20@oswego.edu or Professor Schlegel at daniel.schlegel@oswego.edu. If you have any questions about your rights as a research participant, please contact the Human Subjects Committee Chair, Dr. Theo Rhodes, at hsc-admin@oswego.edu.

Thank you,

Keith Allen | Email: kallen20@oswego.edu

Signature

I have read the above statement about the purpose and nature of the study. I affirm that I am at least 18 years old and I freely consent to participate.

X____________________________
Participant's Signature

X____________________________
Primary Investigator

Participant’s Name:

Principal Investigator:

5.2 Exit Survey Questions

The following are the five questions form the exit survey. Each user had a numbered set of terms assigned to them. The numbers were assigned in the order that users participated and were not tied to the user in anyway. By asking for the user number in the exit survey it ensured that if a user mentioned specific terms, the questions could be matched back the quantitative results.

The remaining four questions offered the user a chance to give their opinions on the system and what it was like to use. The second question asks about the user manual, and what the user liked and disliked about it. Since the DBOL can be tricky to use, and the manual is all the training a user receives, it is important that it can be fine tuned to be more effective. The third questions asks about the general usability of the system. The goal of this question is to find what users struggled with other than the questions, and see if there were suggestions that could be implemented to help future users. The fourth question hoped to question how effective the tip box was. This question may have lead the user to an answer more so than the others, but still we wanted to know more about the users process for answering questions, especially when they were difficult. Finally, the last question is a space for users to voice any questions, comments, or concerns that did not fit any of the other questions.

The results of the user survey were insightful, and while mostly excluded from this paper, are helping to shape future versions of the DBOL, especially in terms of training.

1. What was your user number?
2. Did the DBOL User Manual prepare you for using the system? Was there information or functionality you wished was included, or could be left out?
3. Overall, what did you think of the ease of use for the DBOL? Were there features that were straight forward and helpful? Were there others that were clunky and confusing?
4. If you were unsure of how to answer a question, what would you do to help make your choice? How effective did that help feel and is there another method you would like to see added to assist users?
5. Finally, if you have any other questions, thoughts, ideas, or comments about the system, project, experiment, or this experience, please leave them here.

5.3 DBOL User Manual for Testing

The following three pages are the user manual used for the fall testing. Small tweaks were made between summer and fall testing, but the that manual was omitted due to the changes being covered in section 4.3. The full user manual is not currently available, but changes are continually being made to improve how future DBOL users are trained.

DBOL User Manual for Testing

1 Intro to DBOL

The Dialog Based Ontology Learner is a software tool being developed to assist a user in building a Basic Formal Ontology (BFO) compliant ontology. As the user it is important to know how you can communicate with the DBOL to make the most of your experience. This document will help familiarize yourself with the DBOL.

2 How to use DBOL

2.1 DBOL Layout

The DBOL can be split into four sections. Figure 1.1 shows the DBOL mid-session and has been overlaid with colors to help distinguish the different sections.

2.1.1 Chat Box (Red)

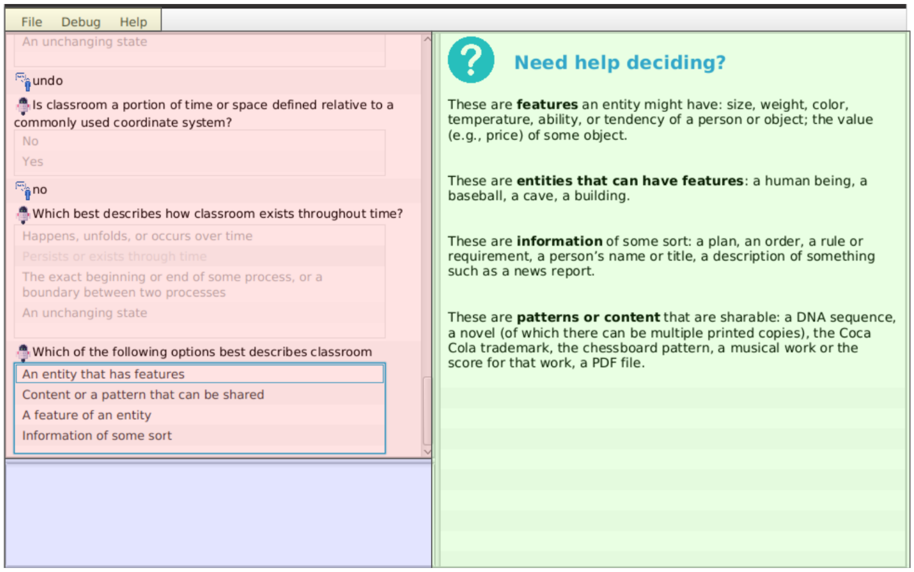
The Chat Box is where the DBOL's questions, answer options, and your answers will appear. Many questions allow for selection from a list, which will be found in the Chat Box. All of the DBOL's messages are marked with a small robot icon and all of your responses are marked with a small human icon. After the first few messages a scroll bar will appear that can be used to look through the past messages of your session.

2.1.2 Text Box (Blue)

The Text Box can be used to type responses to the DBOL. Just type while your cursor is in the text box and hit the `Enter` key to send your text. This text will then appear in the Chat Box.

2.1.3 Tip Box (Green)

Some of the DBOL's questions can be tricky thus the right box shows tips, examples, and non-examples specific to the question currently being asked. If you are having trouble answering a question the tip box might be able to help.



An unchanging state

undo

Is classroom a portion of time or space defined relative to a commonly used coordinate system?

No

Yes

no

Which best describes how classroom exists throughout time?

Happens, unfolds, or occurs over time

Persists or exists through time

The exact beginning or end of some process, or a boundary between two processes

An unchanging state

Which of the following options best describes classroom

An entity that has features

Content or a pattern that can be shared

A feature of an entity

Information of some sort

Need help deciding?

These are **features** an entity might have: size, weight, color, temperature, ability, or tendency of a person or object; the value (e.g., price) of some object.

These are **entities that can have features**: a human being, a baseball, a cave, a building.

These are **information** of some sort: a plan, an order, a rule or requirement, a person's name or title, a description of something such as a news report.

These are **patterns or content** that are sharable: a DNA sequence, a novel (of which there can be multiple printed copies), the Coca Cola trademark, the chessboard pattern, a musical work or the score for that work, a PDF file.

Figure 1: DBOL Layout

2.1.4 Menu Bar (Yellow)

The menu bar has three sections that help you facilitate your session with the DBOL As a user the most important is the `File` drop down which holds the `Load(Replay)Session` , `Save Session`, and `Exit` options. These will be discussed more in the Other Functionality Section.

2.2 Communicating

There are many ways to communicate with the DBOL and many questions can accept answers in multiple formats.

2.2.1 Typing

You can always type your answers to questions. The system is able to correct for small spelling mistakes and incomplete answers. For example, when trying to answer "Every instance is made of matter" typing either "every instance" or "evry instance is made of matter" will select the desired answer.

2.2.2 List Selection

Many of the answers can be long and instead you may want to select your answer from a list (An example can be seen in Figure 1.1 at the bottom of the Chat Box). You can select your desired answer from the list by clicking on it. Unlike typing answers you answer will not be entered into the Chat Box separately, but will become a darker shade of grey (As seen in the middle of the Chat Box in figure 1.1, in which ”Persists or exists through time” was selected).

2.3 Other Functionality

Outside of answering questions there are other functions of the DBOL that as a user you should be aware of

2.3.1 Undo

If you ever want to roll back to answer a question differently just type `undo` into the Text Box. Typing undo rolls back one question, but can be entered many times in a row to reset yourself several questions. Undo moves the system back to that question and forgets any answers from after the point you have rolled back to. This means it is recommended that you roll back as soon as you spot an issue as to limit the number of questions you will answer again for a single term.

Note: In evaluation mode you cannot undo a term assignment. Once a term has been assigned a BFO classification undoing will lead to the previous term being asked about again and the next term being skipped.

However, before a term is assigned the system will ask you to confirm that you want to assign the term. If you do not want to assign the term you must type undo (”No” will not prevent the term assignment).

2.3.2 Exiting

You can exit the system by using either the `Exit` option in the `File` menu, or the `Close` button found in the top right corner. Either option will prompt a pop up asking users if they would like to ”Upload log file to central server for debug purposes?”. Selecting `Yes`, `No`, or closing this menu then closes the DBOL.

3 Notes on Spatial Regions

When we refer to portions of space, this must not be confused with the things which occupy, or are located at, them. Thus, your body occupies a certain portion of space, but can later move to a new region of space. A portion of space in this sense, is also not to be confused with a related usage of the term ’space’. Thus, when we talk of the ”space” inside your nose, your stomach, your car, or a building, we are talking about something different. These, like the objects they are part of, occupy a region of space (itself a smaller spatial part of the space occupied by those objects), and move through space.