Ryan Barker

ECE 4420: Knowledge Engineering

October 29th, 2015

Takehome 2 Report

Section 1: Development of the Jesse 1 Ontology

The development of Jesse1 was fairly straightforward. Classes were developed to represent the life form, mammal, dog (canine), golden retriever, and Jesse concepts (each class in this list was set to the superclass of the next). Since there were no instances in the graphical representation, these classes were set as abstract.

Slots were set up for each relation (consumes, produces, offspring, has, hates, owns, owner, age, and likes). The "consumes" and "has" slots were set for multiple cardinality. For all slots except "has", the default value was set to whatever the relation pointed to in the diagram. Because a life form cannot have a negative age, "age" was restricted to positive numbers. The template values in "has" were set, and then it was overridden at the class level to the appropriate levels for each class. Figure 1 shows the slot representation for Jesse1.

Slot	Type	Facets
consumes	string*	default={food, oxygen}
produces	string	default=waste
offspring	string	default=born live
has	string*	template={warm blood, four legs, water dish, excellent retrieving instinct, heavy fur coat}
hates	string	default=cats
owns	string	default=tennis ball
owner	string	default=Leslie Schalkoff
age	float	default=3.5, minimum=0.0
likes	string	default=riding in cars

Figure 1. Jesse1 slots.

Because there were no instances in Jesse1, completing the class and slot representations completed the ontology. The graphical and HTML representations of Jesse1 are shown in Appendix A.1 and saved in the "Jesse1" subfolder of this archive. The original HTML files can be viewed in the "HTML" subfolder of "Jesse1".

Section 2: Development of the Jesse 2 Ontology

Jesse2 was similar to Jesse1, but, as noted in the PDF, changed "Jesse" to be an instance of "Golden Retriever" rather than a subclass of it and included more specific slots in place of "has". To develop the Jesse2 class hierarchy, the engineer started with the Jesse1 hierarchy,

deleted the Jesse subclass, and changed the “Golden Retriever” subclass to concrete rather than abstract.

The slots for Jesse2 contained all of the slots as Jesse1 except for has and the additional slots shown in the diagram in the specifications. All of the slots shown with the “Jesse” instance still had the necessary restrictions placed on them (namely, age must be greater than zero), but they were not given any default or template values and were instead set in the instance itself. Additionally, a “name” slot was set up to store the name of the Jesse instance and the forms editor was used to display instances of golden retriever by “name”. Figure 2 shows the slots for Jesse2.

Slot	Type	Facets
age	float	minimum=0.0
blood-type	string	default=warm
coat-type	string	default=heavy fur
consumes	string*	default={food, oxygen}
frame-size	string	default=large
has-dish	string	default=water
has-legs	string	default=four
has-owner	string	
hates	string	defualt=cats
likes	string	
name	string	
offspring	string	default=born live
owns	string	
produces	string	default=waste
retrieving-instinct	string	default=excellent

Figure 2. Jesse2 slots.

Jesse2 contained a single instance, “Jesse”. Jesse’s age was set to 3.5, its “has-owner” was set to Leslie Schalkoff, its “like” was set to riding in cars, its “name” was set to Jesse, and its “owns” was set to tennis ball. The graphical and HTML representations of Jesse2 are shown in Appendix A.2 and saved in the “Jesse2” subfolder of this archive. The original HTML files can be viewed in the “HTML” subfolder of “Jesse2”.

Section 3: Final Ontology Topic and Response to Topic Questions

The original inspiration for the final ontology was answering questions of the form, “What audio equipment do I need to purchase to sound like electric guitarist XYZ”? The author of the ontology owns three guitars, is very interested in guitars and guitarists, and wanted a way to quickly answer these questions (as he is asked them far too often). Because answering this question heavily relates to the electrical signal chains guitarists use, the ontology was expanded

to cover the definition of a guitar signal chain with example instances and called “signalchain”. The domain of the ontology is guitars and guitarists and it is a subset of the music knowledge domain.

The sources used for the development of the signalchain ontology were a combination of the author’s own knowledge of the subject matter, data from audio equipment provider websites (Ibanez, Fender, Marshall, etc.), data from reputable audio equipment dealers, and Premier Guitar Magazine “Rig Rundown” videos. In the latter, Rebecca Dirks from Premier Guitar Magazine uses backstage passes to live concerts to meet with guitarists and guitar techs, go over the amplifiers, guitars, and effects units they are using for that show, and report on the experience. Amplifiers, guitars, and effects units also happens to be the way the author represented the initial subclasses of “Audio Equipment”, so these videos worked very well for the author.

The final signalchain ontology developed has a variety of uses via the Protégé queries tab. In the queries tab, the ontology can be queried for either electric guitarists, example signal chains, or audio equipment (guitars, amplifiers, and effects units) by a variety of filters. Among the most significant are the year a guitarist started, the most famous group a guitarist was in, a specific component in an example signal chain, the make and model of pieces of audio equipment, the amplification type of an amplifier, the type of effects units, and various electric guitar characteristics. When an instance of audio equipment is returned from a query, double clicking it will reveal all of its details and a list of guitarists in the ontology who use it. When an instance of a guitarist is returned, double clicking it will reveal all of its details and a concise list of equipment that guitarist uses or will allow a user to sound like that guitarist. When an instance of an example signal chain is returned, double clicking it will reveal the equipment inside of that signal chain.

The intended user of this ontology is any guitarist or aspiring guitarist seeking to gear their sound towards a particular artist they take interest in. The ontology will allow this audience to determine what equipment they will need to purchase to sound like the guitarists represented in it, and the documentation in the ontology will give them a complete picture of all of the characteristics of this equipment and their meanings. The goal is to present these people with the information they need to make smart decisions while purchasing equipment. Further, the documentation and signalchain instances in the ontology are designed to give users an understanding of how to build an electric guitar signal chain once they have equipment. The information in this ontology is also useful to those who are merely curious what equipment certain guitarists use.

The developed ontology does a variety of things, as documented above.

Section 4: Development of Final Ontology

The development of the class hierarchy started with “Audio Equipment”, “Electric Guitarist”, “Guitar Cable”, and “Signal Chain” top level classes. “Audio Equipment” was broken

into “Amplifier”, “Effects Pedal”, and “Electric Guitar” subclasses. “Amplifier” was broken into subclasses by amplification type, “Effects Pedal” was broken into subclasses by effect type, and “Electric Guitar” was broken into subclasses by bridge type. “Electric Guitarist” was broken into “50’s Guitarist”, “60’s Guitarist”, “70’s Guitarist”, “80’s Guitarist”, and “Modern Guitarist” subclasses. For the purposes of this ontology, the term “modern guitarist” was defined as any guitar that started after 1989. “Guitar Cable” was a top level class because cabling instances are necessary for signal chain instances, but guitar cable is not audio equipment and is also not the focus of this ontology. “Signal Chain” did not have any subclasses. Only classes at the most specific levels were marked “concrete”. All others were marked as abstract, since there was a more specific class of that class at the next class below it.

There are numerous slots in the ontology to represent the various characteristics of audio equipment, electric guitarists, guitar cables, and signal chains. This report will only discuss the slots relevant to defining a signal chain and the primary purpose of answering which guitarist uses which equipment. Read the in-Protégé documentation in the ontology for a detailed description of each slot and its associated values.

The “contains_a” and “is_part_of_a” slots are inverses of each other and serve to define what a signal chain is by adding edges between the “Signal Chain” class and its components of “Amplifier”, “Electric Guitar”, “Effects Pedal”, and “Guitar Cable”. This definition is furthered with five example instances, which were all pulled from guitars of the corresponding eras (Sample Chain 1 is the 50’s, Sample Chain 2 is from the 60’s, etc.).

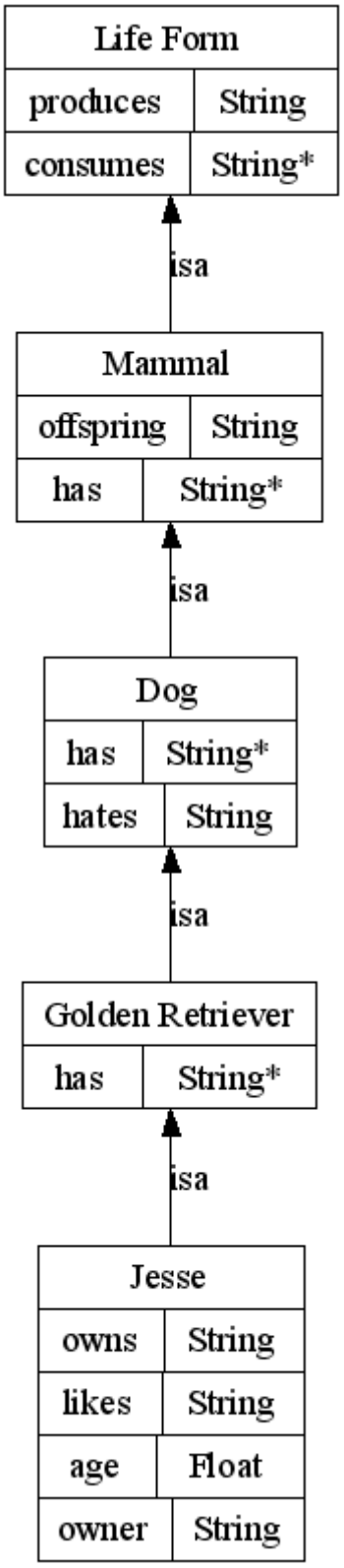
The “uses” and “used_by” slots are inverses of each other and integral to showing which guitarists use which equipment. They are instance list slots (and therefore have cardinality greater than one). The “uses” slot is assigned to all instances electric guitarists to show the instances of audio equipment that they use in the ontology. Similarly, the “used_by” slot is assigned to all instances of audio equipment to show the electric guitarists that use them.

The instances in this ontology were created by representing enough information to cover the equipment for twenty five different electric guitarists and to provide five sample signal chains. There ended up being 121 different instances across the entire file. There was more than one instance of each concrete class, and equipment used coincidentally covered a wide variety of manufacturers.

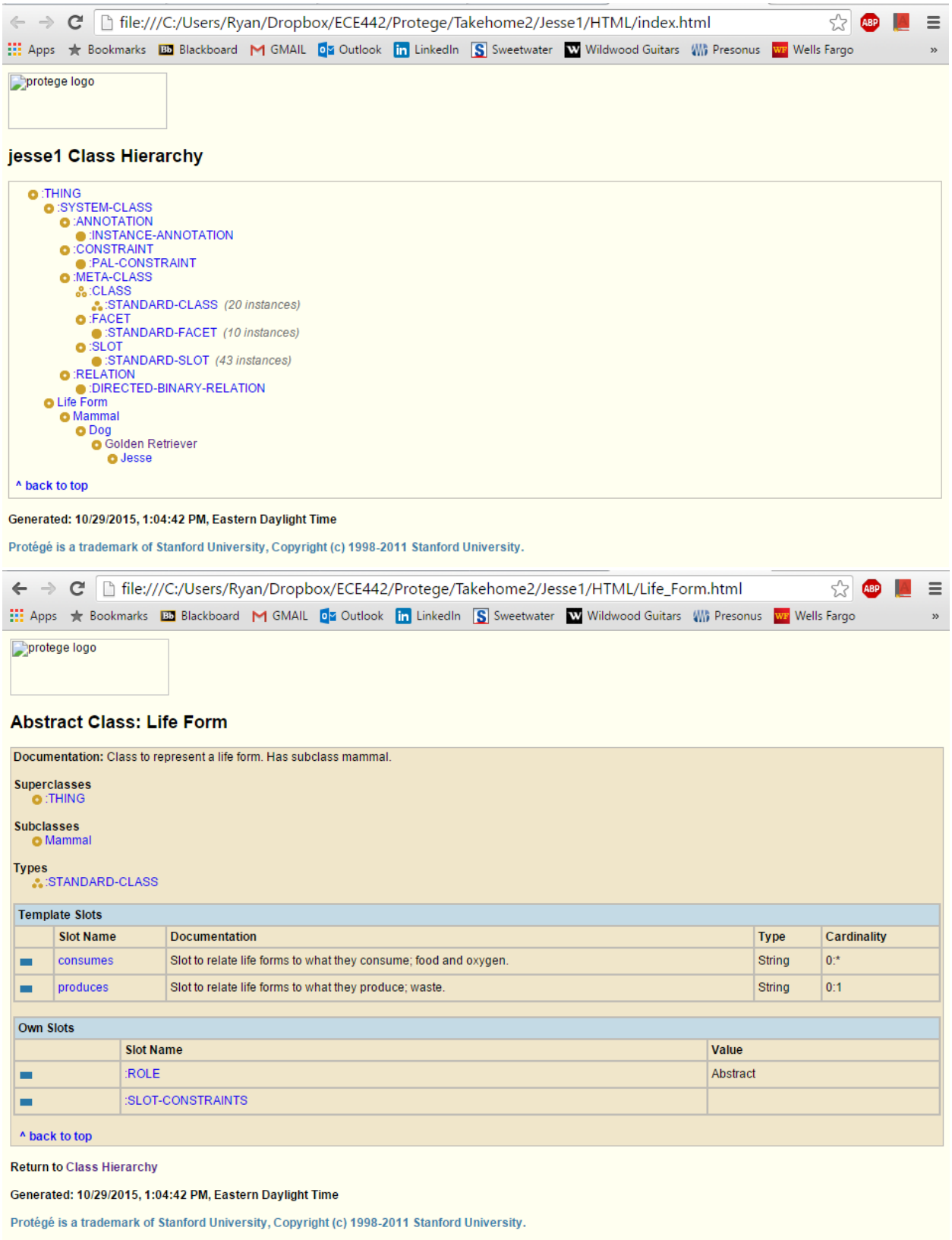
The class and slot hierarchy for the signalchain ontology is shown in Appendix A.3.1.1. A screenshot showing the functions of the “uses” and “used by slots” is shown in Appendix A.3.1.2. The complete graphical representation of signalchain is not shown in this report, but can be viewed in the “signalchain.gif” file in the “SignalChain” directory. This directory also contains other .gif files to show the ontology at various levels. Appendix A.3.2 shows the HTML representation of SignalChain. The original HTML files can be viewed in the “HTML” subfolder of “SignalChain”.

Appendix

A.1.1: Jesse1 Graphical Representation



A.1.2: Jesse1 HTML Screenshots



file:///C:/Users/Ryan/Dropbox/ECE442/Protege/Takehome2/Jesse1/HTML/index.html

Apps Bookmarks Blackboard GMAIL Outlook LinkedIn Sweetwater Wildwood Guitars Presonus Wells Fargo

protege logo

jesse1 Class Hierarchy

- :THING
  - :SYSTEM-CLASS
    - :ANNOTATION
      - :INSTANCE-ANNOTATION
    - :CONSTRAINT
      - :PAL-CONSTRAINT
    - :META-CLASS
      - :CLASS
        - :STANDARD-CLASS *(20 instances)*
      - :FACET
        - :STANDARD-FACET *(10 instances)*
      - :SLOT
        - :STANDARD-SLOT *(43 instances)*
    - :RELATION
      - :DIRECTED-BINARY-RELATION
  - Life Form
    - Mammal
      - Dog
        - Golden Retriever
          - Jesse

^ back to top

Generated: 10/29/2015, 1:04:42 PM, Eastern Daylight Time

Protégé is a trademark of Stanford University, Copyright (c) 1998-2011 Stanford University.

file:///C:/Users/Ryan/Dropbox/ECE442/Protege/Takehome2/Jesse1/HTML/Life_Form.html

Apps Bookmarks Blackboard GMAIL Outlook LinkedIn Sweetwater Wildwood Guitars Presonus Wells Fargo

protege logo

Abstract Class: Life Form

Documentation: Class to represent a life form. Has subclass mammal.

Superclasses
- :THING

Subclasses
- Mammal

Types
- :STANDARD-CLASS

Template Slots

	Slot Name	Documentation	Type	Cardinality
▬	consumes	Slot to relate life forms to what they consume; food and oxygen.	String	0:*
▬	produces	Slot to relate life forms to what they produce; waste.	String	0:1

Own Slots

	Slot Name	Value
▬	:ROLE	Abstract
▬	:SLOT-CONSTRAINTS	

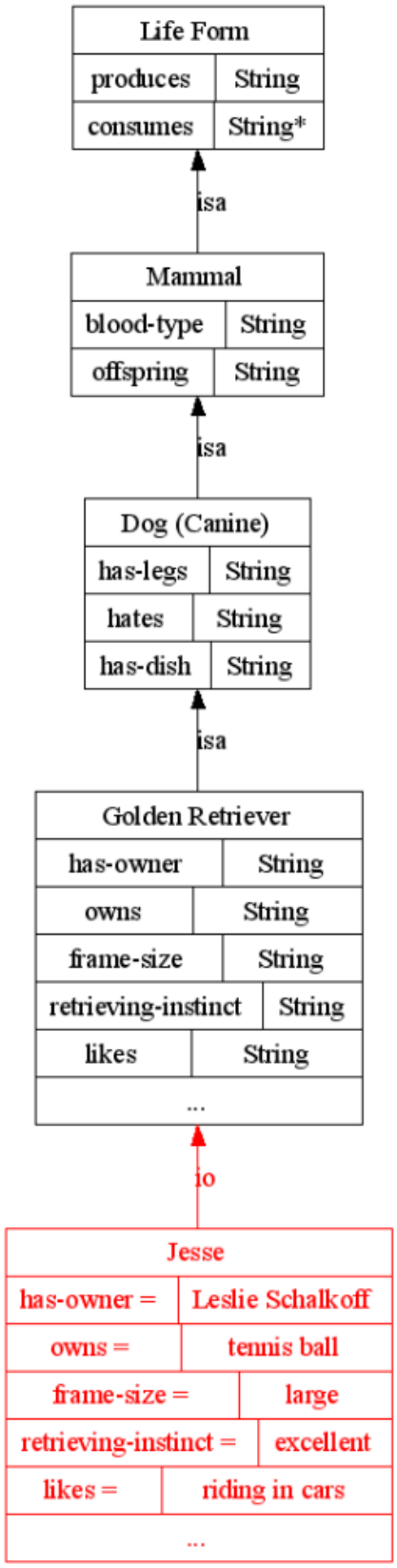
^ back to top

Return to Class Hierarchy

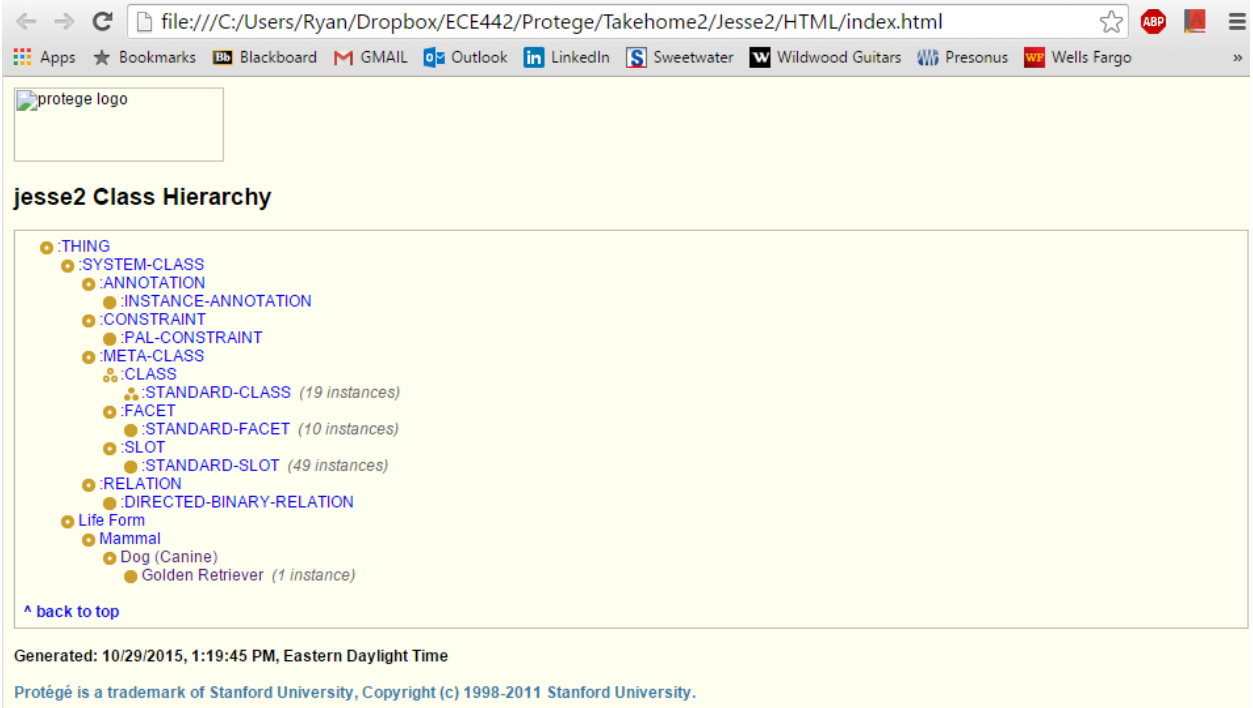
Generated: 10/29/2015, 1:04:42 PM, Eastern Daylight Time

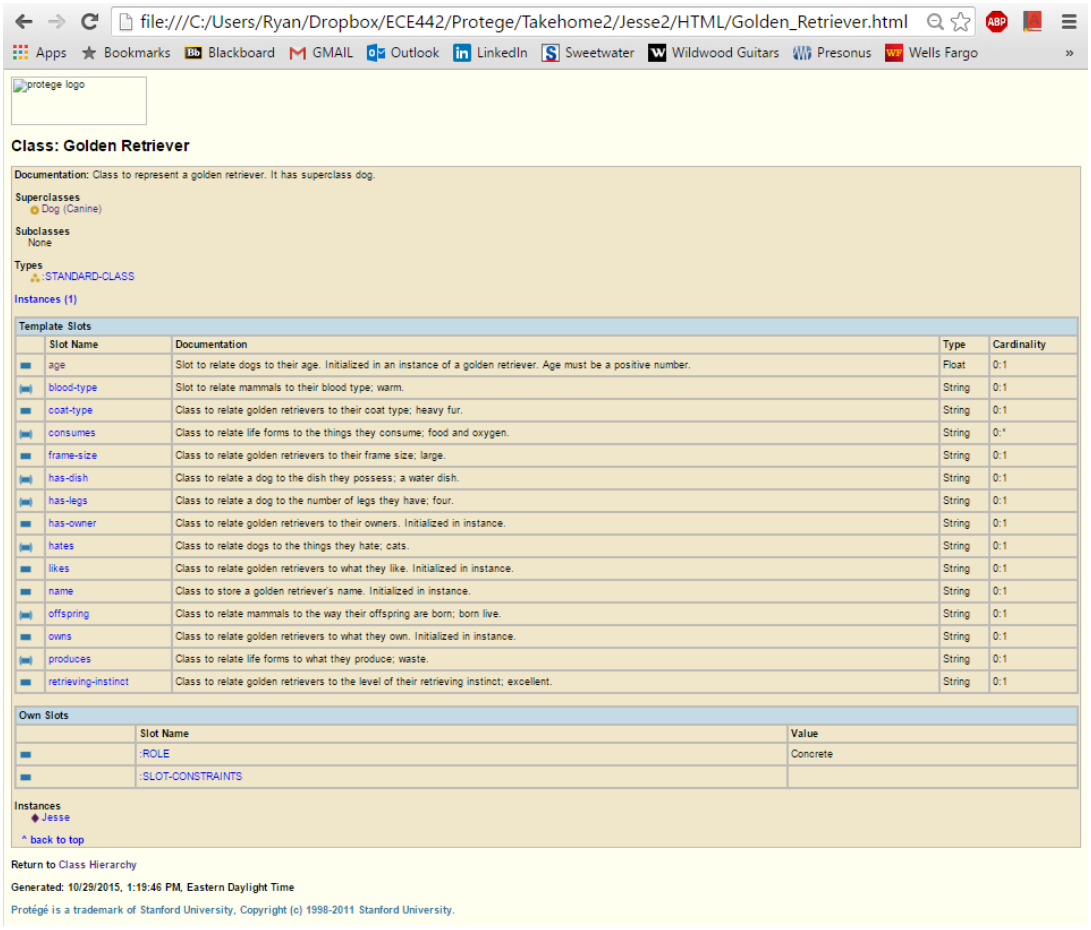
Protégé is a trademark of Stanford University, Copyright (c) 1998-2011 Stanford University.

A.2.1: Jesse2 Graphical Representation

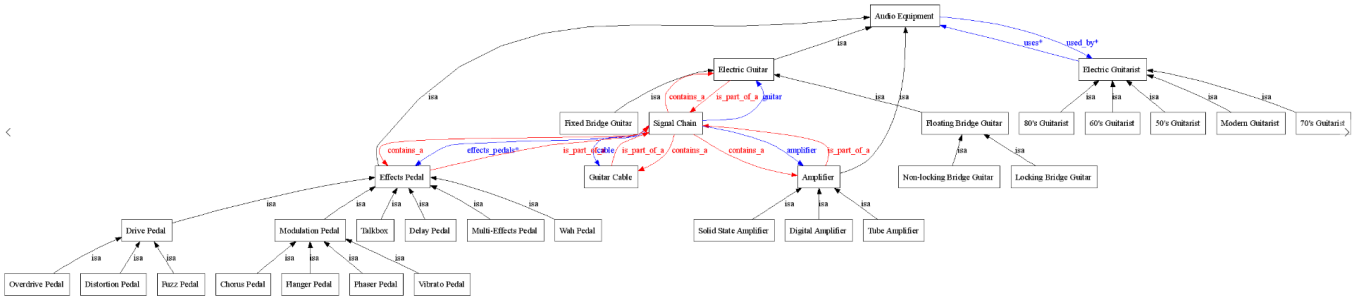


A.2.2: Jesse 2 HTML Screenshots

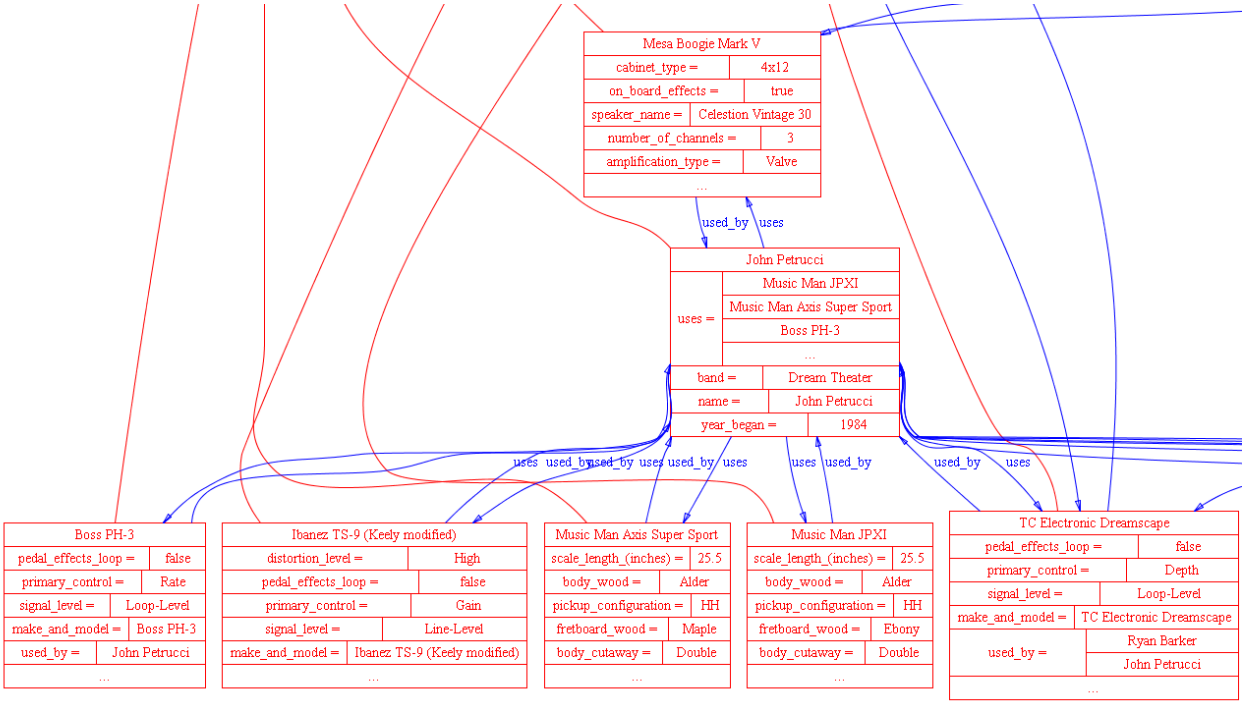




A.3.1.1: SignalChain Class and Class Edge Hierarchy



A.3.1.2: SignalChain Sample Electric Guitarist/Audio Equipment Relations



A.3.2: SignalChain HTML Screenshots

protege logo

Class: Tube Amplifier

Documentation: Class for all amplifiers utilizing vacuum tube amplification. Because tube amps are seperated into heads and cabinets, this class contains an extra slot for cabinet type that the other amp classes do not share. Most artists use 4x12 cabinets. Tube amplifiers do not typically contain effects units, so "on board effects" is set false.

Superclasses
- Amplifier

Subclasses
None

Types
- :STANDARD-CLASS

Instances (17)

Template Slots

Slot Name	Documentation	Type	Cardinality
amplification_type	Slot to represent the method of amplification used by guitar amplifiers as one of three symbols. Symbols: - Valve: Vacuum Tube based amplifiers that work with true analog signal. These include most traditional amplifiers and are still the most popular type of amplifier today. - Solid State: Semiconductor-circuit based amplifiers that work with true analog signal. Solid state amplifiers are less expensive alternatives to tube amplifiers. - Digital: Amplifiers that use a digital processor to amplify guitar signal. These are either cabinet-based or rack-based.	{Valve, Solid-State, Digital}	0:1
cabinet_type	Slot used to describe the cabinet type of a tube amplifier with symbols. Symbols: - 1x12 Cabinet: A cabinet containing one, twelve inch diameter speaker for amplification. - 2x12 Cabinet: A cabinet containing two, twelve inch diameter speakers for amplification. - 4x12 Cabinet: A cabinet containing four, twelve inch diameter speakers for amplification.	{1x12, 2x12, 4x12}	0:1
is_part_of_a	Slot used as a class relation to show what makes up a signal chain along with the "contains a" relation. Useful to show users how to build a signal chain out of an artist's gear.	Class	0:1
make_and_model	A string used to describe the model number and manufacturer of a piece of audio equipment. This is used as the label for all instances of audio equipment.	String	1:1
number_of_channels	An integer slot describing the number of channels contained in an amplifier. To function, all amplifiers must have at least one channel, so this value must be at least one. Definition: An amplifier channel is a section of an amplifier dedicated to producing either a clean or distorted sound. When an amp is switched on, only one channel can be active at a time.	Integer	0:1
on_board_effects	A boolean slot to describe whether an amplifier is equipped with on-board effects. Definition: Some amplifiers have built in processors with on-board effects. These can include but are not limited to reverberation, chorus, delay, and flange. Amplifiers with on-board effects are typically digital.	Boolean	0:1
power_rating_(watts)	An integer slot used to show the power rating of an amplifier. All amplifiers are rated for at least one watt, so this value must be greater than or equal to one.	Integer	0:1
speaker_name	A slot used to specify the speaker's used in an amplifier's cabinet as a string. Not applicable to rack units.	String	0:1
unique_features	A string slot with multiple degrees of cardinality. This is used to list any miscallaneous features of audio equipment not described by the other slots, but is limited to five entries to reduce clutter.	String	0:5
used_by	Relation used to show which electric guitarists use a particular piece of audio equipment. One half of linking famous artists to the musical equipment they use.	Electric Guitarist	0:*

Own Slots

Slot Name	Value
:ROLE	Concrete
:SLOT-CONSTRAINTS	

Instances
- Blackstar Artisan 100
- Blackstar HT-5
- Carvin V-3M
- Carvin VL-300 Legacy 3
- EVH 5150 III
- Fender '65 Twin Reverb
- Fender ML-212 Hot Rod Deville
- Jet City 20
- Marshall JCM-800
- Marshall JVM-410
- Marshall Plexi
- Marshall YJM-100
- Mesa Boogie King Snake
- Mesa Boogie Mark V
- Orange AD-30HTC
- Suhr PT-100
- Two Rock TR-JMS

^ back to top

Return to Class Hierarchy

Generated: 10/26/2015, 7:22:20 PM, Eastern Daylight Time