

README

Daniel Hadad

November 17, 2022

1 Introduction

Hi there! I'm someone who is very passionate about a variety of strategy games, most of which feature many complex elements. I'm also trying to teach myself semantic web technology, so I thought it would be fun to start building ontologies for some of my favourite games. This ontology is for the fantastic board game Agricola by Uwe Rosenberg.

This document describes the elements of this repository, along with some of my thoughts on them. I'm very new at this, so if you see something that doesn't make sense or could be obviously improved, please don't hesitate to email me with suggestions!

1.1 Project Goals

The goal of this project is two-fold: first, I want to create a searchable card database for Agricola, including conceptual information about cards like their strategic uses and potential synergies; and second to create a searchable database of played games of agricola from the play-agricola website.¹

The eventual hope is that a website with a functional UI can be built to allow users to execute card searches, and that data analysis can be performed on the game data, for both competitive and game development purposes.

2 Structure of the Data Architecture

2.1 The Ontology

The card and game databases are constructed from a single ontology using OWL/RDF. The ontology provides a representation for all of the physical game elements, as well as relevant abstract concepts. In addition, the ontology represents the elements pertaining to instances of played games of Agricola, such as which players participated in the game, and the particular decisions made during each turn of the game. Thus, this single ontology is a sufficient basis for triplestores of available cards as well as a detailed log of played games.

In order to achieve this, the base ontology includes a number of named individuals that are necessary for representing various game states, concepts, and card information. As a matter of best practice I endeavored to keep the ontology free of any named individuals, but decided that it was better to include a (relatively) small number that would be shared among the different triplestores.

As it stands, the specifics of the ontology will not be easy to parse for anyone who is not very familiar with the game rules and concepts. I think it would be nice to eventually write a document that facilitated governance of the ontology for someone who hasn't played the game, but for right now I want to improve my skills with actually making the ontology in the first place!

¹In the future I hope it will be possible to import game data from other online platforms as well, but the initial scope of the project is restricted to play-agricola.

2.2 The Card Database

The card database is a triplestore of information available about the various cards in Agricola, including conceptual features not physically present on the cards, such as their strategic utility, synergies with other cards, and relative strength.

2.3 The Game Replay Database

Full logs of played games can be recorded as triples. When ready, there will be a python script available in this repository that parses log files from the play-agricola website and transforms relevant game information into triples. This allows game logs to be represented in a consistent, general way that lends itself to much more efficient data analysis than has been previously available to the community. The ability to use SPARQL queries to perform data analysis allows the competitive and development communities to ask questions with greater control of granularity. Furthermore, the replay and card databases can work in conjunction to allow even more interesting data analysis.

3 Other Files in this Repository

There are a number of files that I've been tinkering with for demonstration/refinement purposes. I'll briefly describe them here.

3.1 Cost Triples

The folder 'cost_triples_example' contains a spreadsheet file and transformation rules for the Cellfie plugin for Protégé. The 'costs' are named individuals in the base ontology that each have a value associated with a physical game element and an integer amount.² These individuals are a work around for the restriction to 2-place relations in OWL. If you know of a more efficient way to do this, please let me know!

3.2 Card Data

The 'CardData' folder contains a few different files that I used to create a test demo of the card data triplestore.

3.2.1 Demo CSV

The 'EIK_demo_info' file is a .csv file containing some card database information for the cards in the original version of the game. Much of the information was gathered using Eric Ho's spreadsheet that I edited. I plan to eventually release a version of this spreadsheet that is organized in a more optimal way for adding to the triplestore.

3.2.2 Shortcut Expansion

In order to make it easier to input multiple categories for cards into the spreadsheet, I created a number of shortened input codes. The 'sed_shortcut_expansion' file can be run using the bash shell 'sed' command to expand those codes into their respective ontology URIs. I will eventually make a python script to accomplish this same task.

3.2.3 The triples

The other files are the SPARQL queries that actually construct the card data triplestore. 'construct_card_data_triples' uses Tarql to construct triples based on the .csv file.

'cleanup_triples' is a SPARQL query I made to work around my own limited expertise with Tarql. Many of the .csv columns have multiple values separated by semi-colons. Tarql allows for column-splitting when constructing triples, but I was only able to do this for all the columns at once using multiple construct clauses. As a result, the triples for some named individuals end up 'scattered' throughout the ontology file, rather than bundled together. This makes it cumbersome to read through the .ttl file when not using a program to view the ontology.

²**N.B.** the 'Cost' prefix is not semantically relevant in terms of gameplay. For instance, if a player receives two wood as a result of taking an action, this would be represented as a triple of the form 'SomePlayer receives Cost2Wood'.

I used Apache Jena's ARQ command line tool with the 'cleanup triples' file to gather all of a named individual's triples together.