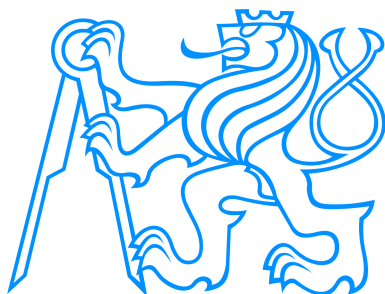


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ



DIPLOMOVÁ PRÁCE

Ontologie a jejich využití v geoinformatice

Praha, 2008

Tereza ROGALEWICZOVÁ

Příliš žlutoučký kůň úpěl ďábelské ódy.
PŘÍLIŠ ŽLUŤOUČKÝ KŮŇ ÚPĚL ĎÁBELSKÉ ÓDY.

Prohlášení

Prohlašuji, že jsem tuto práci vypracovala samostatně, pouze s použitím podkladů uvedených v příloženém seznamu.

V Praze dne 19. prosince 2008

Poděkování

Dovoluji si na tomto místě poděkovat vedoucímu diplomové práce Ing. Jiřímu Cajthamlovi, Ph.D. za uvedení do tématiky a prohlédnutí rukopisu. Současně připojuji poděkování všem, kteří mi poskytli podklady pro tuto práci.

Zadání diplomové práce.

Abstrakt

Ontologické inženýrství je nová, v současné době se rozvíjející oblast informatiky. Na světové úrovni proběhlo nebo probíhá již několik projektů zaměřených právě na tuto problematiku. V České republice je výzkum teprve na začátku.

Diplomová práce přináší základní informace o problému sémantického zpracování informací a o ontologiích, jako o jednom ze způsobů řešení zachycení sémantiky. Dále poskytuje přehled možných forem formálního zápisu ontologií a softwaru, který lze pro tyto účely použít.

Z výsledků práce lze nahlédnout, jaké jsou možnosti využití ontologií v oblasti informatiky s důrazem na použití v geoinformatice a kartografii.

Klíčová slova – Keywords

ontologické inženýrství – ontology engineering
sémantika – semantics
ontologie – ontology
geoinformatika – geoinformatics

Ontology engineering is a new area of informatics that has developed in the last time. Several projects focused on this field have been already carried out abroad. However, the research is still in its infancy in the Czech Republic.

The dissertation brings basic information on the problem of semantic processing of information, and on ontologies as a solution of semantic rendering. Furthermore, it offers an overview of possible forms of formal notations of ontologies and software that can be used for this purpose.

The results of the dissertation show possibilities for utilization of ontologies in the field of informatics, when the accent is on utilization in geoinformatics and cartography.

Obsah

Úvod	1
1 Sémantický web	2
1.1 Sémantika	3
1.2 Metadata	5
1.3 Identifikace zdrojů	6
2 Ontologie	9
2.1 Filosofické kořeny	9
2.2 Pojem ontologie	10
2.3 Členění ontologií	10
2.4 Struktura ontologií	12
3 Jazyky	14
3.1 Historické ontologické jazyky	14
3.2 Historické „webové“ ontologické jazyky	16
3.3 Moderní „webové“ ontologické jazyky	18
3.4 Deskripční logika	19
4 RDF	21
4.1 Syntaxe RDF	21
4.2 RDF grafy	24
5 RDF Schéma	27
5.1 Třídy	27
5.2 Vlastnosti	29

6	OWL	31
6.1	Verze jazyka OWL	31
6.2	Syntaxe	32
6.3	Struktura OWL dokumentu	32
6.4	Vlastnosti	37
7	Editory ontologií	40
7.1	CMapTools Ontology Editor	42
7.2	DERI Ontology Management Environment	42
7.3	Chimaera	42
7.4	Java Ontology Editor	43
7.5	Karlsruhe ontology	43
7.6	Karlsruhe ontology 2	43
7.7	Knoodl	44
7.8	Ontolingua Ontology Editor	44
7.9	Synaptica	44
7.10	Swoop	45
7.11	TopBraid Composer	45
8	Editor Protégé	46
8.1	Instalace	46
8.2	The Protégé-Frames Editor	47
8.3	The Protégé-OWL Editor	47
8.4	Protégé Users	47
8.5	Uživatelské rozhraní	48
9	Návrh ontologií	53
9.1	Stanovení rozsahu a cíle ontologie	53
9.2	Identifikace entit specifických v dané doméně	54
9.3	Uspořádání entit do hierarchie	54
9.4	Definice entit	54
9.5	Vlastnosti entit	54
9.6	Identifikace vztahů	55
9.7	Výběr vhodného softwaru	55

10 Ontologie a jejich využití	56
10.1 Ontologie v geoinformatice, kartografii a geodezii	56
10.2 Návrh ontologií pro geoinformatiku, kartografii a geodezii	57
11 Závěr	59
Literatura	60
Seznam zkratk	61
Seznam tabulek	63
Seznam obrázků	64
Příloha A	65
Příloha B	66

Úvod

Ontologické inženýrství je jednou z moderních oblastí informatiky. Souvisí s rozvojem internetových služeb v komerční sféře a se snahou o efektivnější využití Internetu nejen jako prostředku k výměně informací srozumitelných pro člověka, ale i pro počítače.

V současné době nelze využívat sémantických znalostí ukrytých v textech na webových stránkách. Odpovědí na tento problém má být v budoucnu přechod od normálního webu (dnes tzv. webu 2. generace) k webu sémantickému. Zachycení a vyjádření sémantiky je realizováno pomocí matedat, jejichž nejvyšší formou jsou právě ontologie.

Co jsou ontologie, jak ontologie vytvářet a kde lze ontologie využívat jsou otázky, na něž se tato diplomová práce pokusí odpovědět.

Kapitola 1

Sémantický web

Otázka budoucího možného nástupu sémantického webu vyplývá ze stavu webu současného. S rozšiřováním internetových služeb a jejich dostupnosti pro širokou veřejnost se na webu stále zvětšuje množství informací. S tím ovšem souvisí i problém, jak tyto informace efektivně vyhledávat a třídit. Stávající způsob založený na textovém vyhledávání nebo vyhledávání pomocí klíčových slov začíná být nedostačující. Je pravdou, že současný web a technologie určené k jeho tvorbě se stále vyvíjí. Jedná se ale pouze o oblast formátu výstupu informací, které jsou prezentovány uživateli. Získané informací musí však uživatel stále sám vyhodnocovat a na základě toho pak případně pokračovat ve hledání. Tento problém by však mohl vyřešit právě sémantický web.

Myšlenku sémantického webu poprvé prezentoval ve své vizi koncem 90. let 20. století Tim Berners-Lee (tvůrce současného webu, ředitel konsorcia W3C¹). V květnu roku 2001 spolu s dalšími spolupracovníky v článku v Scientific American [1] důrazně upozornil na skutečnost, že současná síť WWW je v podstatě pouze velké množství webových stránek, které neustále roste a kde je stále složitější nalézt požadované relevantní informace. Východiskem z tohoto chaosu má být postupný přechod od stávajícího webu k webu sémantickému, který je, jak T. Berners-Lee zdůrazňoval, rozšířením současného webu, jež datům přiřazuje přesný význam.

Sémantický web by tedy prezentoval informace takovým způsobem, aby byly srozumitelné jak pro člověka, tak pro stroje. Pod pojmem „informace srozumitelná pro stroje“ je myšlena schopnost speciálních softwarových aplikací vyhledat a zpracovat obsah informace získané z webu. Obrazně řečeno, by tyto aplikace měly „vědět“, co nalezený obsah znamená a co s ním dále dělat. Na dosažení tohoto cíle existují dva pohledy.

1. *Zanesení sémantiky do webových stránek.* První možností je rozšíření informací prezentovaných na webu o informace sémantické. Toto tzv. „sémantické značkování“ by znamenalo přidání jistého typu metadat, jež by poskytla formální sémantiku obsahu webové stránky. To by ale znamenalo vytvoření obecně platné specifikace sémantiky všech informací, které by pak speciální softwarové aplikace byly schopné porozumět a

¹<http://www.w3.org>

mohli tak obsah dále automaticky zpracovat. Jedním z hlavních problémů tohoto řešení je právě ona vhodná, obecně platná reprezentace sémantiky.

2. *Přidání sémantiky do aplikací.* Druhou alternativou zanesení sémantiky do webu je její přímé zapracování do aplikací. Tyto aplikace by pak dokázaly informace klasickými metodami vyhledat a na základě znalostí informace zpracovat. Na tomto principu zpracování informací pracují například webové aplikace, označované jako „obchodní agenti“. Tito „obchodní agenti“ dokáží vyhledat informace z webu na základě vybraných klíčových slov a nalézt tak například nejlevnější cenu požadovaného výrobku.

1.1 Sémantika

Sémantika je nauka o významu slov, morfémů a jiných znaků a jejich vztahu ke skutečnosti, kterou označují. Samotný pojem sémantika vznikl z řeckého slova *séma*, které znamená význam.

Jak již bylo řečeno, právě sémantika, a tedy i chápání významu obsahu různých webových zdrojů, bude základním kamenem sémantického webu. Podle vztahu sémantiky k sémantickému webu ji můžeme rozdělit do čtyř skupin. Následující rozdělení vychází z článku [2].

- Implicitní.
- Neformální (explicitní).
- Formální sémantika pro zpracování člověkem.
- Formální sémantika pro strojové zpracování.

1.1.1 Implicitní sémantika

V nejjednodušším případě lze sémantiku chápat jako implicitní. Význam v takovém případě vychází ze společného porozumění lidí danému termínu. Obecně známým příkladem tohoto druhu sémantiky je používání XML, jako je cena, adresa, datum, jméno aj.

V samotném XML dokumentu, DTD (Document Type Definition), nebo XML schématu není nikde specifikován význam těchto značek. Nicméně, existuje-li obecně platný konsensus o významu těchto značek, pak lze jejich sémantiku zapracovat do webových aplikací. Na tomto principu fungují, již dříve zmiňovaní, „obchodní agenti“.

Nevýhoda implicitní sémantiky spočívá právě v mnohoznačnosti. Nejsou určena žádná pravidla pro zachycení sémantiky a tak může často docházet k nedorozumění. Například i zdánlivě jednoduchý termín „cena“ může mít v různých implementacích různé významy (různá měna, zahrnutí daně, apod.).

1.1.2 Neformální sémantika

Dalším přístupem je neformální, neboli explicitní sémantika. Jedná se o sémantiku vyjádřenou neformálním způsobem a tedy určenou především pro zpracování člověkem.

Příkladem takového druhu sémantiky je slovník cizích slov. Pro všechna hesla je zde neformálně, přirozeným jazykem vyjádřen jejich význam. Ovšem vzhledem ke složitosti přirozeného jazyka je možnost přímého strojového zpracování značně omezená.

1.1.3 Formální sémantika pro zpracování člověkem

I v tomto případě se jedná o sémantiku explicitní. Hlavním rozdílem od předchozích typů sémantik je její vyjádření pomocí formálního jazyka. Ovšem i přesto se jedná o sémantiku určenou pouze pro zpracování člověkem. Lze si ji představit jako formální dokumentaci či jako formální specifikaci významu.

1.1.4 Formální sémantika pro strojové zpracování

Explicitní a formálně vyjádřená sémantika je určená pro stroje a umožňuje přímé zpracování včetně automatické inference. Hlavním předpokladem je, že pokud aplikace narazí na nový, doposud nedefinovaný pojem, bude schopna pojem automaticky zpracovat a případně o něm něco odvodit. Ovšem, jak tohoto dosáhnout, zůstává stále problémem.

K uvedenému problému můžeme přistoupit dvěma způsoby, a to procedurálně nebo deklarativně. První přístup, přístup procedurální, spojuje sémantiku se spuštěním určité procedury ve chvíli, kdy aplikace narazí na určitý příznak. Druhý, deklarativní přístup zachycuje sémantiku pomocí formální deklarace. Tento způsob se jeví jako vhodnější pro sémantický web. Ovšem velkým úskalím tohoto způsobu je, že nemůže fungovat zcela obecně. Je zapotřebí rozumět symbolům a znát syntaktická pravidla použitého jazyka. Bez těchto znalostí by se jednalo o komplikovanou kryptografickou úlohu, která je obtížně řešitelná pro člověka, natož pak pro stroj.

Abychom přešli výše zmíněnému problému, je třeba vycházet z určitých předpokladů. Následující část vychází a cituje z [2].

1. *Stejný jazyk reprezentace.* Různé ontologické jazyky vychází z různých paradigmat, mají různou vyjadřovací sílu, různou míru formální podpory pro vyjádření sémantiky, různou míru schopnosti inference atd. Z těchto důvodů je nutné, aby byl obsah napsán jazykem, který je aplikaci znám.
2. *Logicky kompatibilní konceptualizace.* Stejný jazyk nezaručuje „dorozumění“ dvou stran.
3. *Veřejně deklarované postupy.* Ani při sdíleném jazyku a kompatibilní konceptualizaci nelze vyloučit, že dva různí lidé nebudou pro stejnou doménu používat různé ontologie. Dva různé pojmy mohou mít stejný význam a stejný pojem může mít dva různé významy. Tentýž koncept může být modelován s různou mírou podrobnosti. Určitá myšlenka může být vyjádřena s využitím rozdílných jazykových primitivů.

1.2 Metadata

Pojem metadata začal být předmětem zájmu odborné veřejnosti v 90. letech 20. století. Se vznikem a rozvojem digitálních knihoven, databází a katalogů vyvstala otázka, jakým způsobem efektivně popisovat datové záznamy. Řešením byla právě metadata.

Samotné slovo metadata vzniklo spojením řeckého slova *meta*, v překladu mezi, a latinského slova *data*, čili „to, co je dáno“. Samotných definicí existuje celá řada. Liší se od sebe především v závislosti na tom, jak budou dále metadata využívána.

Definice jsou například následující, podle [2]:

- Data o datech.
- Informace o informacích.
- Význam nebo sémantika dat.
- Zdroj poskytující informace o jiném zdroji.
- Popisná informace o webovém zdroji.

Kromě tohoto ovšem také platí, že některá metadata mohou být zároveň metadaty, ale i daty samotnými. Je zřejmé, že pouhé obecné vymezení „data o datech“ je pravdivé, ale nedostačující. V souvislosti se sémantickým webem je třeba uvést definici T. Berners-Lee: „Metadata jsou stroji srozumitelné informace o webových zdrojích nebo dalších věcech.“

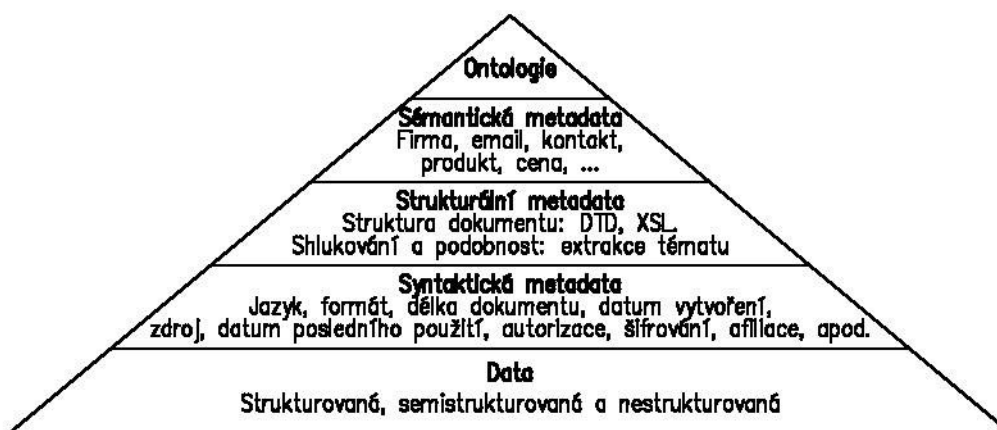
Hlavním předpokladem všech definicí je, že metadata jsou stále jen data, z čehož vyplývá, že mohou být součástí nějakého zdroje. Mohou obsahovat informace o sobě samém, o dalších zdrojích, o způsobu uložení dat a jejich správě, o jejich významu, způsobu využití nebo shrnutí obsahu dat. Dále mohou obsahovat i informace nepřímo související s obsahem dat, jako je například jméno autora dokumentu, datum jeho vytvoření atd. Použití metadat s sebou přináší velkou výhodu. Tou je redukce množství informací díky těmto vlastnostem, podle [2]:

1. Umožňují abstrahovat od detailů reprezentace a zachycují informační obsah nezávisle na původní formě dat.
2. Umožňují reprezentovat doménové znalosti popisem informační oblasti, k níž přísluší výchozí data. Tato znalost může následně posloužit k domněnkám o daných datech.

Metadata lze klasifikovat podle různých kategorií. V následujícím textu bude uvedena nejčastější klasifikace metadat podle úrovně abstrakce, s níž metadata popisují obsah dokumentu. Následující část cituje z [1].

1. *Syntaktická metadata* se soustřeďují na podrobnosti o zdroji dat. Tento druh metadat je vhodný pro katalogizaci nebo kategorizaci.
2. *Strukturální metadata* se zaměřují na strukturu dokumentu jako takového. Tyto informace o dokumentu lze využít při ukládání, zpracování nebo prezentaci, jako navigace a zjednodušení vyhledávání informací. Typickým příkladem je XML schéma.

3. *Sémantická metadata* popisují kontextově relevantní informace, přičemž se soustřeďují na doménově specifické elementy založené na ontologii, která je srozumitelná uživatelům orientujícím se v dané doméně. Tímto způsobem lze dosáhnout smysluplné interpretace dat a vysoké interoperability.
4. *Ontologie* představují nejvyšší formu metadat a současně jsou klíčovým principem sémantického webu. (Více v následujícím textu)



Obrázek 1.1: Typy metadat

1.3 Identifikace zdrojů

1.3.1 URL

URL², Uniform Resource Locator („jednotný lokátor zdrojů“), je řetězec znaků s danou strukturou, který slouží k přesné specifikaci umístění zdrojů informací na Internetu. Zápis URL definuje doménovou adresu serveru, umístění zdroje na serveru a protokol, kterým je možné zdroj zpřístupnit.

Ukázka struktury URL:

`http://www.server.cz/dokumenty/doc.html`

Protokol: http

Server: www.server.cz

Zdroj: /dokumenty/doc.html

²http://en.wikipedia.org/wiki/Uniform_Resource_Locator

Zdroj je uveden včetně cesty v rámci webserveru, začínající od adresáře „viděného“ z Internetu. Je zřejmé, že prostřednictvím URL lze identifikovat pouze zdroje, jenž jsou umístěny na Internetu.

1.3.2 URN

URN³, Uniform Resource Name („jednotný název zdroje“), označuje název zdroje, pomocí něhož lze zdroj identifikovat nezávisle na jeho umístění. Na URN je kladeno několik podmínek, jejichž splnění je nutné pro funkčnost URN. Patří sem:

1. *Zachování významu názvu v globálním měřítku.* Název nesmí představovat konkrétní umístění.
2. *Jedinečnost.* Stejně URN nesmí být přiřazeno více různým zdrojům.
3. *Stálá životnost.* Předpokládá se, že význam URN bude trvalý a navždy jedinečný. Vždy je třeba brát ohled na životnost zdroje.

RFC⁴ 2141 popisuje v BNF⁵ syntaxi URN takto:

$$\langle \text{URN} \rangle ::= \text{"urn:"} \langle \text{NID} \rangle \text{" :"} \langle \text{NSS} \rangle$$

Výraz $\langle \text{NID} \rangle$, Namespace Identifier, představuje identifikátor jmenného prostoru. Druhý výraz, výraz $\langle \text{NSS} \rangle$, Namespace Specific String, představuje specifický řetězec jmenného prostoru. Fráze uzavřené v uvozovkách jsou povinné.

Následující příklad URN ukazuje zápis pro knihu The Last Unicorn autora Petera S. Beagla:

urn:isbn:0451450523

1.3.3 URI

URI⁶, Uniform Resource Identifier („jednotný identifikátor zdrojů“), je řetězec znaků s danou strukturou, který slouží k přesné specifikaci zdroje informací na Internetu. Neomezuje se pouze na umístění nebo na název zdroje jako URL a URN, ale je obecnější. URI může popisovat zdroj jak čistě z hlediska jeho identity bez určení umístění, tak čistě z hlediska toho, kde je možné zdroj nalézt, bez popisu jeho identity. Zdroj lze ovšem rovněž specifikovat na základě jeho identity i jeho umístění. Pomocí URI je samozřejmě možné identifikovat zdroje pomocí Internetu dosažitelné i nedosažitelné. Jelikož je samotné URI velmi obecný koncept, je i jeho formát velmi volný:

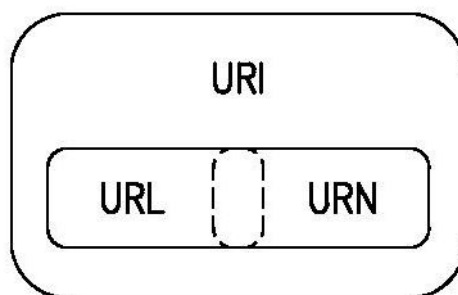
schéma : libovolný řetězec znaků, jehož význam i syntax závisí na použitém schématu

³http://en.wikipedia.org/wiki/Uniform_Resource_Name

⁴http://en.wikipedia.org/wiki/Request_for_Comments

⁵http://cs.wikipedia.org/wiki/Backus-Naurova_forma

⁶http://en.wikipedia.org/wiki/Uniform_Resource_Identifier



Obrázek 1.2: Vennův diagram vztahu URI, URN, URL

1.3.4 URI reference

URI reference je dalším typem řetězce, který může URI reprezentovat, neboli zdroj identifikovaný URI. URI reference může představovat plnou URI, v tom případě hovoříme o absolutní URI referenci, nebo jen její část specifickou pro dané schéma. Pak jde o URI referenci relativní.

Absolutní URI reference identifikuje zdroj bez ohledu na kontext, ve kterém se URI reference nachází. Oproti tomu relativní URI reference je v podstatě zkrácenou formou URI reference absolutní. K URI referenci lze navíc připojit tzv. identifikátor fragmentu, oddělený od zbytku URI znakem `#`. Pomocí tohoto identifikátoru je pak možné zpřesnit identifikaci zdroje nebo některé jeho části.

Absolutní URI reference:

`http://www.server.cz/dokumenty/doc.html#odstavec1`

Relativní URI reference:

`/dokumenty/doc.html#odstavec1`

Kapitola 2

Ontologie

2.1 Filosofické kořeny

Ontologie je pojem pocházející z filozofie 18. století, který můžeme vyložit jako nauku o bytí, nebo také Univerzální soustavu znalostí popisující jevy, zákonitosti a objekty světa. První ontologické koncepty můžeme ovšem nalézt už u před Sokratovských myslitelů. Zde se povětšinou omezovaly na snahu redukovat mnohotvárnost smyslového světa na jeden základní princip (*arché*). Významný pokrok nastal až v myšlení Platóna a posléze jeho žáka Aristotela. Aristoteles se ve svých spisech *Metafyzika* a *Kategorie* nesnaží o pouhou redukci veškeré skutečnosti na jeden základní princip, ale snaží se rozpoznat základní vlastnosti a rozčlenění celé skutečnosti. Ve své koncepci světa rozlišuje deset kategorií, jež měly vyčerpat všechny základní třídy toho, co může být. Zmíněné kategorie jsou tyto: substance (konkrétní objekt), kvantita, kvalita, relace, kde, kdy, poloha, mít, činnost a trpnost. Další alternativní klasifikační systém se objevuje až ve 13. století, kdy se Ramon Llull pokusil vypracovat logicko-kombinatorický grafický systém klasifikace veškerého poznání, nezávislý na jazyku, náboženství a kultuře.

Samotný výraz ontologie se objevuje až v pracích německých filosofů R. Gockela, J. Lorharda a Ch. Wolffa. I. Kant používá výraz ontologie místo tradičního termínu metafyzika ve smyslu nauky o jsoucnu a bytí jako takovém. Posléze se objevuje pojem formální nebo deskriptivní ontologie pro označení zkoumání otázek klasifikační povahy (jaké existují třídy jsoucen, jaké jsou mezi nimi vztahy).

V oblasti informatiky se výraz ontologie pravděpodobně poprvé objevuje v práci J. McCarthyho. V 80. letech 20. století dává ontologie do souvislosti s informatikou J. F. Sowa. V roce 1993 se objevuje první definice ontologie T. Grubera, který definuje ontologii jako *explicitní specifikaci konceptualizace*. Tato definice byla poté modifikována W. Borstem, podle něhož je ontologie *formální specifikace sdílené konceptualizace*. Z této definice se vychází dodnes. Následně se objevují ještě další definice, z nichž můžeme citovat například definici J. F. Sowy z roku 2000:

„Předmětem ontologie je studium kategorií věcí, které existují nebo mohou existovat v určité doméně. Výsledkem tohoto studia je katalog věcí, jejichž existenci předpokládáme v dané doméně D, z perspektivy osoby používající jazyk L, aby mluvila o D.“

2.2 Pojem ontologie

V následujícím textu budeme vycházet z modifikované definice T. Grubera: „*Ontologie je formální, explicitní specifikace sdílené konceptualizace*“. Další část cituje z [3].

Konceptualizace říká, že se jedná o takový abstraktní model výseku reálného světa, který identifikuje relevantní koncepty daného výseku. Explicitní znamená, že je jednoznačně definován typ konceptu i podmínky jeho použití. Formální poukazuje na to, že ontologie by měla být strojově zpracovatelná. Je možná různá míra formality. Sdílený odráží představu, že ontologie zachycuje konsensuální znalosti, tj. že není omezena na jedince, ale je akceptována šířeji.

Z definice vyplývá, že ontologie je určitým systémem zachycení reality, jenž lze přenášet a sdílet. Cílem ontologického inženýrství je tedy vytvořit formální systém k zachycování reality jako celku, jejích určitých oblastí, odvozování v rámci těchto oblastí, modelování, dotazování, atd.

Za základní způsoby využití ontologií jsou obvykle považovány tyto, podle [4]:

- Podpora porozumění mezi lidmi.
- Podpora komunikace (interoperability) mezi počítačovými systémy.
- Základ pro návrh znalostně-orientovaných aplikací.

Dalším pojmem, se kterým se můžeme v souvislosti s ontologiemi setkat, jsou ontologické závazky. Ontologické závazky (ontological commitment) jsou jisté základní teoretické předpoklady týkající se přístupu k modelování reality, z nichž daná ontologie vychází a které může explicitně či implicitně vyjadřovat. Při práci s ontologiemi je nutné brát toto na vědomí, jelikož tento fakt částečně omezuje případnou integraci s ontologiemi nesoucími jiné ontologické závazky. Na základě tohoto pak rozlišujeme několik typů ontologií.

2.3 Členění ontologií

Jak již bylo dříve řečeno, ontologie je souhrnným označením pro systém popisující realitu. Z tohoto důvodu ji lze rozčlenit podle několika hledisek.

2.3.1 Členění podle oborových oblastí

V závislosti na historickém vývoji vlivu ontologií na jednotlivé obory, můžeme ontologie rozdělit na následující:

1. *Terminologické* (lexikální) ontologie lze ztotožnit s pokročilými tezaury (slovníky) užívanými v oborech orientovaných na textové zdroje. Hlavním charakteristickým rysem je ústřední role termínů, které nejsou ale dále nijak definovány, a jejich vzájemné relace.
2. *Informační* ontologie jsou rozvinutím databázových konceptuálních schémat. Hrají roli nadstavby nad primárními zdroji, pro které zajišťují konceptuální abstrakci a vyšší úroveň kontroly integrity.
3. *Znalostní* ontologie navazují na výzkum v oblasti reprezentace znalostí v rámci umělé inteligence. Ontologie jsou chápány jako logické teorie s relativně volnou vazbou na reálné objekty. Třídy a relace jsou definovány pomocí formálního jazyka.

2.3.2 Členění podle míry formalizace

Přestože jednou z hlavních vlastností ontologií je formalizace, nalézejí v praxi své uplatnění i ontologie semi-formální či zcela neformální. Jedná se zpravidla o glosáře, v nichž jsou jednotlivé pojmy vysvětleny přirozeným jazykem. Ontologie vyjádřené pomocí formálních jazyků, můžeme dále dělit podle formálně-logických vlastností daného jazyka, jako je úplnost a rozhodnutelnost. Ovšem přesné hranice zde nejsou. Ve většině případů bývá formální ontologie doplněna o dokumentaci, jež je vyjádřena přirozeným jazykem.

2.3.3 Členění podle předmětu formalizace

Pravděpodobně nejčastějším způsobem členění ontologií je právě členění podle předmětu formalizace (zdroje konceptualizace). Mezi základní typy patří:

1. *Doménové* ontologie jsou nejčastěji užívaným typem ontologií. Předmětem je vždy určitá specifická věcná oblast, šířeji, či úžeji vymezená.
2. *Generické* ontologie (někdy označované jako ontologie vyššího řádu) naopak usilují o zachycení obecných zákonitostí, které platí napříč nejrůznějšími věcnými oblastmi. Někdy se ještě v rámci generických ontologií vyčleňují tzv. ontologie vyšší úrovně a ontologie typu common-sense. Ontologie vyšší úrovně („upper-level“) usilují o co nejobecnější zachycení pojmů a vztahů jako základu taxonometrické struktury každé další ontologie. Ontologie typu common-sence („přirozeného rozumu“) naopak obsahují řadu velmi specifických, avšak relativně doménově nezávislých znalostí. Generické ontologie bývají často považovány za základ struktury dalších typů ontologií.
3. *Úlohové* ontologie (reprezentační, metaontologie) označují generické modely znalostních úloh a metod jejich řešení. Na rozdíl od ostatních ontologií se zaměřují na procesy odvozování. Tento typ ontologií bývá využíván pro modely znalostních úloh a metody jejich řešení. Mezi úlohy tradičně řešené pomocí těchto znalostních modelů lze počítat diagnostiku, plánování, zhodnocení apod.
4. *Aplikační* ontologie jsou nejspecifičtější. Jedná se o konglomerát modelů adaptovaných na konkrétní aplikaci, zahrnující zpravidla doménovou i úlohovou část.

2.4 Struktura ontologií

Přestože je základní struktura ontologií prakticky ve všech projektech, jazycích i nástrojích stejná, v těchto používaná terminologie se od sebe značně liší. Tyto odlišnosti často vedou k nejrůznějším problémům. Asi největší rozdíly panují mezi tradičními jazyky pro tvorbu ontologií, jako je například Ontolingua, a jazyky novými, především webovými. Následující část popisující struktury ontologií a rozdíly v terminologii čerpá z [3].

2.4.1 Třídy, koncepty, kategorie, rámce

Základním stavebním kamenem ontologií jsou třídy, které označují hierarchicky uspořádané množiny konkrétních objektů. Termínu třída odpovídá v některých formalismech termín koncept (neboli pojem), popřípadě kategorie, a úzce souvisí i s termínem rámec jako základním konstruktem mnoha systémů umělé inteligence.

Termín „ontologické“ třídy na rozdíl od tříd v objektově-orientovaných modelech neobsahuje procedurální metody. Naopak je spíše definován jako unární relace. Třídy, které mají specifikovány podmínky nutnosti a postačitelnosti, se označují jako definované, ostatní třídy pak jako primitivní. Pokud se jedná o celou množinu tříd je vždy definována hierarchie mezi jednotlivými třídami.

2.4.2 Individua, instance

Individuum odpovídá konkrétnímu objektu reálného světa. Je tedy do jisté míry protikladem třídy. Individua patří do třídy objektů, které mohou být označeny pomocí třídy. Ovšem individuum může být do ontologie vloženo i bez příslušnosti k nějaké třídě. Avšak tato možnost nemusí být některými ontologickými jazyky podporována. Termín instance je chápán jako ekvivalentní, avšak často asociuje příslušnost k určité třídě, což nemusí být nutně skutečností. Instance třídy označuje vlastně člena třídy.

Je ovšem třeba si uvědomit, že ontologie prvotně slouží k popsání konceptů (tříd) a nikoli faktů o konkrétních objektech. Proto některé jazyky individua nepodporují. Je-li třeba individua využít, pracují s ním jako s atomickou třídou.

2.4.3 Relace, funkce, sloty, vlastnosti, role, atributy

Obdobně jako je tomu u databázových modelů, jsou podstatnou složkou ontologií vztahy čili relace n -tic objektů. V tradičních jazycích mohou být pojmenované relace specifikovány pomocí libovolných logických podmínek. V některých případech jim lze pouze přiřadit řada předdefinovaných omezení. Pro binární relace, tedy relace pouze mezi dvěma objekty, se používá označení sloty nebo vlastnosti. Takové relace potom definují výhradně vztahy mezi jednotlivými objekty a nejsou nijak svázány s konkrétními třídami.

Jako zvláštní typ relace jsou chápány funkce. V souladu s běžným matematickým chápáním jde o n -ární relace, u nichž je hodnota n -tého argumentu jednoznačně určena

předchozími ($n - 1$). Funkční slot je označován jako atribut a předpokládá se, že je definován pro všechny instance třídy.

2.4.4 Meta-slots, omezení na slots, facets

V ontologiích je možné, na rozdíl od jiných systémů, přiřazovat slotům (relacím) vlastnosti. Takové slots pak označujeme jako meta-slots. Nejčastějším meta-slotem je právě hierarchický vztah podřízeného a nadřazeného slotu. Ten můžeme chápat jako tranzitivní binární meta-relaci. Symetrickou binární meta-relací je vztah slotu vůči slotu k němu inverznímu. Ostatní meta-slots vesměs odpovídají unárním relacím (symetrie, tranzitivita, atd.).

Vedle obecných matematických vlastností patří mezi vlastnosti slotů také definiční obor (domain) a obor hodnot (range). Uvedené vlastnosti můžeme označit jako globální omezení, jelikož se vztahují ke slotu bez ohledu na způsob jeho použití. Lokální omezení na slotech (slot constraints, property restrictions), neboli facets, vymezují hodnoty slotu aplikovaného na konkrétní třídu ze svého definičního oboru. Jedná se zejména o omezení kardinality nebo oboru hodnot slotu.

2.4.5 Primitivní hodnoty a datové typy

Argumenty relací nemusí být jen n -tice objektů, jako je tomu u tzv. objektových slotů. Mohou jimi být i primitivní hodnoty, které žádnému objektu neodpovídají. V to případě hovoříme o tzv. dato-typových (datatype) slotech. Obor hodnot dato-typového slotu bývá vymezen některým základním datovým typem (string, integer, ...), číselným intervalem nebo výčtem. Někdy se primitivní datové hodnoty označují jako dato-typové instance.

2.4.6 Axiomy, pravidla

Vedle výrazů explicitně vymezujících příslušnost ke třídám a relacím je obvykle možné do ontologií zařazovat další logické, výrokové formule. Tyto se nejčastěji označují jako axiomy. Axiomy mohou mít v ontologii zcela samostatné postavení (např. v OIL) nebo jsou syntakticky zařazeny do definice tříd a relací (např. v DAML+OIL), samy však vůči nim nemají definiční roli. V „pragmatických jazycích“ (např. OntoBroker) se spíše označují jako pravidla a mají i tomu odpovídající sémantiku.

2.4.7 Souhrnné informace

Vedle těchto znalostních konstruktů nesou ontologie ještě další doplňující informace, obvykle umísťované do hlavičky. Tyto tzv. souhrnné informace zahrnují zejména odkazy na importované ontologie, dokumentační položku s údaji o autorovi, verzi, času a způsobu vytvoření apod. Je zřejmé, že údaje obsažené v souhrnných informacích nemají žádný význam pro logickou interpretaci ontologie.

Kapitola 3

Jazyky

Nezbytným předpokladem pro strojové zpracování informací je zachycení struktury dat. Toho je možné dosáhnout několika způsoby. Prvním způsobem, se kterým se setkáváme například u databází relačního typu, je vyjádření struktury dat pomocí tabulek s atributy. V prostředí Internetu, kde se setkáváme především s dokumenty textového charakteru, se uplatňuje tzv. značkování dokumentů. Pod tímto značkováním se rozumí, že ([3]) „určité znakové sekvence obsahují informaci, která připisuje obsahu dokumentu určitou roli.“ Tyto značky mohou mít různou podobu, ale nejčastěji se setkáváme s podobou slov uzavřených do lomených závorek, tagy, jako je tomu kupříkladu u značkovacího jazyka HTML¹ nebo XML². Značkovací jazyk tedy definuje sadu tagů, jež mohou být využity, dále jak mohou být vzájemně kombinovány a jaký nesou jednotlivé tagy význam. Některé značkovací jazyky (např. XML) navíc nabízí možnost definovat nové, specifické tagy v závislosti na potřebách konkrétní aplikace.

3.1 Historické ontologické jazyky

Potřeba formálně popisovat a reprezentovat ontologie vyústila ve vznik několika ontologických jazyků v relativně krátkém období. Patřily mezi ně jazyky jako je Cyc, Ontolingua, OCML nebo OKBC a XOL.

V dalším textu budou tyto jazyky popsány.

3.1.1 Cyc

Prvním z pokusů o zachycení znalostí o světě ve velkém rozsahu je stále „živý“ projekt Cyc³, jehož název je odvozen od slova *enCYClopedia*. Projekt byl zahájen v roce 1984 pod vedením D. Lenata. Cílem bylo shromažďovat všeobecné znalosti, které by ve znalostních systémech doplňovaly znalosti expertní, a zabráňovaly absurdnímu chování. Ačkoli se jedná

¹<http://www.w3.org/MarkUp>

²<http://www.w3.org/XML>

³<http://www.cyc.com>

o projekt jednoznačně zaměřený na oblast ontologií, tento termín autoři nepoužívají. Namísto toho se hovoří o „mikroteoriích“ složených z dílčích tvrzení.

Pro formální reprezentaci využívá Cyc vlastní jazyk CycL. Základní notace byla převzata z funkcionálního jazyka LISP⁴. Jazyk má plnou vyjadřovací sílu predikátového kalkulu v kombinaci s prvky rámcových jazyků. Zpočátku se projekt Cyc orientoval pouze na uzavřené komerční aplikace, nabízené firmou CyCorp, a zveřejněna byla pouze nepatrná část znalostí, označovaná jako Upper Cyc Ontology. Průlom nastal až v roce 2001, kdy došlo k úplnému zveřejnění. OpenCyc (Open Source Version of the Cyc(r) technology) je v současné době jednou z největších a nejkompletnějších znalostníchází a zároveň i nástrojem pro smysluplné usuzování. Často je používán jako podklad při vytváření inteligentních aplikací. Poslední verze je verze 1.0.2.

3.1.2 Ontolingua

Dalším pokusem byla iniciativa na začátku 90. let 20. století vedená T. Gruberem a jeho spolupracovníky ze standfordské Knowledge System Laboratory (KSL)⁵. Od počátku se jednalo o veřejný projekt. Hlavním cílem bylo vyvinout dostatečně mocný a přehledný jazyk, jenž by umožňoval sdílet ontologie v rámci odborných komunit používajících vzájemně nekompatibilní znalostní systémy. Výsledkem těchto snah se stal jazyk Ontolingua⁶, který byl koncipován jako nadstavba jazyka KIF⁷ (Knowledge Interchange Format). Jedná se opět o variantu predikátového kalkulu využívající syntaxe LISP. Základními stavebními prvky jazyka Ontolingua jsou definice tříd, relací a funkcí. Vymezující podmínky pro definování příslušnosti instancí jsou vyjádřeny právě v KIFu.

Ontolingua nebyla od počátku koncipována jako „klasický“ jazyk, ale jako mezi-jazyk. Primárně byla určena k výměně ontologických informací mezi systémy, které interně používají vlastní způsob reprezentace ontologií. To však vedlo k jistému omezení možností odvozování přímo v tomto jazyce. Na druhou stranu značná rozšířenost vedla k tomu, že po řadu let plnila Ontolingua roli jazyka „první volby“ pro tvorbu ontologií nezávisle na konkrétním znalostním systému. Důležitou roli pro zpřístupnění ontologií v jazyce Ontolingua hrál tzv. Ontolingua server, umístěný na KSL.

3.1.3 OCML

Reakcí na omezené možnosti definování v jazyce Ontolingua byl jazyk OCML⁸. OCML, neboli Operational Conceptual Modelling Language, navrhl E. Mottu z Open University ve Velké Británii jako jazyk, jenž by (při zachování ontologické podstaty) výrazněji podporoval přímý vývoj programových aplikací, aniž by bylo nutné model překládat do jiného jazyka.

⁴[http://en.wikipedia.org/wiki/Lisp_\(programming_language\)](http://en.wikipedia.org/wiki/Lisp_(programming_language))

⁵<http://ksl.stanford.edu>

⁶<http://www.ksl.stanford.edu/software/ontolingua>

⁷<http://ksl.stanford.edu/knowledge-sharing/kif>

⁸<http://kmi.open.ac.uk/technologies/ocml>

Dalším významným pokrokem bylo těsné propojení vývoje jazyka s tvorbou jeho interpreta, který byl implementován v prostředí CommonLISP.

„Základem interpreta jsou algoritmy pro proglovské dokazování a dědění v hierarchii tříd; třídy a jejich atributy jsou ovšem důsledně chápány jako unární resp. binární relace, takže primární (vnitřní) reprezentací jsou Hornovy klauzule.“ ([4]) Co se týče „deklarativní“ části OCML, byla navržena ve shodě s jazykem Ontolingua. Navíc je však podporována řada konstruktů převzatých z procedurálních jazyků a expertních systémů. K dispozici je i možnost volání LISPovských funkcí. Díky tomu lze v OCML napsat prakticky libovolnou aplikaci, která nemusí mít s ontologiemi ani nic společného.

Jazyk OCML si v řadách odborníků na znalostní inženýrství získal značný respekt. Často bývá označován jako „operacionální Ontolingua“. Skutečností ale zůstává, že se příliš nerozšířil mimo Open University a její projektové partnery.

3.1.4 OKBC a XOL

V roce 1993 začal vznikat na základě analogie s ODBC návrh aplikačního rozhraní Open Knowledge Base Connectivity (OKBC⁹). OKBC umožňuje otevřenou komunikaci mezi rámcově-orientovanými znalostními systémy. Tato komunikace je možná díky protokolu, který specifikuje způsob předávání konstruktů znalostních bází (tj. třídy, individua, sloty) a volání operací nad těmito konstrukty. Vlastní znalostní model OKBS vychází z řady již existujících znalostních systémů a používaná sémantika odpovídá souboru konstruktů ontologických jazyků jako je Ontolingua.

V roce 1999 se jednoduchá verze OKBC, označovaná OKBC-Lite, stala základem pro návrh dalšího ontologického jazyka, eXtensible Ontology Language (XOL¹⁰). Hlavním důvodem vzniku tohoto jazyka byla potřeba bioinformatické komunity sdílet struktury znalostí o výzkumu v oblasti genů. Hlavní a zároveň i nejzásadnější inovací oproti předchozím jazykům bylo použití syntaxe XML, jež umožnila využití i řady obecných nástrojů již existujících pro tento značkovací jazyk. Nad rozdíl od XML Schématu zavádí XOL pouze jednu generickou definici typu dokumentu (DTD), která definuje strukturu tříd, jejich sloty (pouze binární), obory hodnot těchto slotů atd. Jména tříd, atributů atd. neodpovídají názvům, nýbrž hodnotám prvků XML.

3.2 Historické „webové“ ontologické jazyky

3.2.1 SHOE

První jazyk vzniklý s potřeby dodání sémantiky k webovým stránkám byl SHOE¹¹ (Simple HTML Ontology Extension), který v roce 1996 vyvinul tým J. Hendlera na University of Maryland. Tento jazyk umožňuje vkládat do HTML kódu webových stránek:

⁹<http://www.ai.sri.com/okbc>

¹⁰<http://www.ai.sri.com/pkarp/xol>

¹¹<http://www.cs.umd.edu/projects/plus/SHOE>

- Metadata o objektech, jichž se stránky týkají.
- Vlastní ontologie, jež definují sémantiku metadat.

Jazyk SHOE dovoluje, narozdíl od pozdějších „webových“ jazyků, definovat relace s libovolnou aritou. Dále je rovněž možné zapisovat inferenční pravidla ve formě zjednodušených Hornových klauzulích prvního řádu.

3.2.2 Ontobroker

Přibližně v téže době, jako vznikl jazyk SHOE, se na univerzitě v Karlsruhe začíná rodit koncepce jazyka Ontobroker¹². Zakladní myšlenkou je, podobně jako u jazyka SHOE, obohacení webových stránek o sémantické anotace. Zásadní rozdíl je v architektuře obou jazyků, jenž je v případě Ontobrokeru důsledně centralizovaná. Z toho následně vyplývá i předpoklad existence ontologického serveru, který by spravoval všechny ontologie a k němuž by měli přístup pouze vybraní uživatelé. Dalším významným prvkem je i samotné získávání dat pro potřeby ontologií. V tomto případě se nejedná o náhodné prohledávání webu, ale spíše o cílené návštěvy stránek registrovaných poskytovatelů informací z odborných komunit. Důsledkem výše zmíněných požadavků je, že Ontobroker nepoužívá jednotný jazyk pro zachycení anotací a ontologií. Zatímco jazyk anotací je v podstatě pouze jednoduché obohacení HTML o dodatečné atributy, jazykem ontologií je propracovaný formalismus označovaný F-logic¹³. Jedná se o jeden z řady logických kalkulů vytvořených pro potřeby rámcových systémů (F v názvu je zkratkou pro „*Frame*“).

3.2.3 DAML-ONT, OIL

V polovině roku 2000 byl zahájen projekt DAML¹⁴ (DARPA Agent Mark-up Language). Projekt byl sponzorován vojenskou institucí DARPA¹⁵ a od jeho počátku se na vývoji podílela celá řada výzkumníků sémantického webu. Cílem bylo vytvořit sémantický jazyk pro formát RDF, který by měl větší vyjadřovací sílu než RDFS. Jazyk DAML-ONT vychází z předchozích výzkumů v oblasti ontologických jazyků. Obsahuje řadu konstruktů pro vymezení vztahů tříd a hodnot slotů.

Koncem 90. let 20. století dospěl celosvětový výzkum ontologií k závěru, že bude nové jazyky potřeba postavit na některém z propracovanějších logických kalkulů, jenž by umožňoval konstrukci složitých podmínek při zachování dobrých vlastností pro výpočty. Vybrána byla deskripční logika (podrobněji v kapitole 3.4), která se stala základem jazyka OIL¹⁶ (Ontology Inference Layer).

¹²<http://www.ontoprise.de>

¹³<http://en.wikipedia.org/wiki/F-logic>

¹⁴<http://www.daml.org>

¹⁵<http://www.darpa.mil>

¹⁶<http://www.ontoknowledge.org/oil>

3.3 Moderní „webové“ ontologické jazyky

3.3.1 DAML+OIL

Sloučením jazyků DAML-ONT a OIL vznikl jazyk s názvem DAML+OIL¹⁷ (DARPA Agent Mark-up Language + Ontology Inference Layer). Základem jsou třídy reprezentované buď přímo pomocí URI nebo pomocí logických výrazů. Složitější výrazy lze tvořit pomocí konstruktů, které můžeme libovolně kombinovat. Seznam konstruktů je uveden v následující tabulce.

Konstruktor	Popis
intersectionOf unionOf complementOf	Booleovské výrazy
oneOf	Množina primitivních hodnot
toClass hasClass	Existenční nebo universální omezení na slot (třídy, jež jsou hodnotami slotu)
hasValue	Omezení na konkrétní hodnotu
minCardinalityQ maxCardinalityQ CardinalityQ	Omezení na kardinalitu slotu

Tabulka 3.1: Konstruktory logických výrazů

Vlastní náplň ontologií pak tvoří axiomy, vytvořené nad výrazy reprezentujícími třídy. Seznam axiomů je uveden v následující tabulce.

Axiom	Popis
subClassOf	Vztah nadřazené a podřazené třídy
sameClassAs	Ekvivalence tříd
disjointWith	Prázdný průnik tříd
subPropertyOf	Vztah nadřazeného a podřazeného slotu
samePropertyAs	Ekvivalence vlastností
inverseOf	Vzájemně inverzní vlastnosti
transitiveProperty	Tranzitivita vlastnosti
uniqueProperty	Vlastnost je funkcí
unambiguousProperty	Inverzní vlastnost k dané vlastnosti je funkcí
sameIndividualAs	Totožnost individuí
differentIndividualFrom	Odlišnost individuí

Tabulka 3.2: Axiomy jazyka DAML+OIL

¹⁷<http://www.w3.org/TR/daml+oil-reference>

3.4 Deskripční logika

Deskripční logika¹⁸, neboli logika pojmů (Description logics, DL), je pojem zastřešující celou řadu příbuzných logických formalismů, pomocí nichž lze reprezentovat znalosti. Jejich společným rysem je snaha o zachycení struktury tříd a relací obdobně, jako je tomu v případě ontologií. Od klasických ontologických jazyků se liší především tím, že nepředpokládají apriorní vymezení vztahu třídy a podtřídy (taxonomie). Vztahy mezi jednotlivými třídami (subsumpce tříd) jsou vyhodnocovány na základě jejich popisů (deskripcí), jež jsou reprezentovány pomocí různě složitých logických výrazů. Operace prováděné nad těmito výrazy jsou následující, citováno z [5]:

- Zjišťování vztahů (nadřazenosti, podřazenosti) mezi třídami.
- Testování konzistence modelu, tj. splnitelnosti formulí vymezujících třídy.
- Zjišťování příslušnosti instancí ke třídám.

3.4.1 Syntaxe deskripční logiky

Teorie v deskripční logice (označována rovněž jako „terminologie“) odpovídá konečné množině axiomů. Tyto jsou konstruovány pomocí konceptů (tříd), rolí (binárních relací) a atributů. Pojmenované koncepty, role a atributy pak odpovídají atomickým výrazům, z nichž lze skládat další, složitější výrazy. Citováno z [5].

- *Koncepty* odpovídají třídám a jsou vymezené konceptuálními výrazy („concept expressions“). Jsou tedy interpretovány jako množiny objektů.
- *Role* odpovídají relacím a jsou vymezené relačními výrazy („role expressions“). Jsou interpretovány jako binární relace na objektech.
- *Atributy* odpovídají funkcím a jsou vymezeny funkčními výrazy („attribute expressions“).
- *Výrazy* jsou tvořeny pomocí konstruktorů.

3.4.2 Sémantika

Podle [5]. Výrazy mají množinovou sémantiku nad universem instancí Δ^I . Interpretační funkce $\cdot^I$ mapuje každý konceptuální výraz C na podmnožinu C^I universa Δ^I , každý relační výraz R na funkci s množinovými hodnotami (resp. binární relaci) R^I nad universem Δ^I a každý funkční výraz A na parciální funkci A^I s definičním oborem $\text{dom}(A^I) \subseteq \Delta^I$. Sémantika složených výrazů je odvozena ze sémantiky jejich složek.

¹⁸http://en.wikipedia.org/wiki/Description_logic

Konstruktor	Syntaxe
Konjunkce	$C \sqcap D$
Disjunkce	$C \sqcup D$
Negace	$\neg C$
Existenční omezení	$\exists R.C$
Univerzální omezení	$\forall R.C$
Omezení počtu (kardinalita)	$(\leq n R), (\geq n R)$
Existenční omezení atributu	$\exists A.C$
Univerzální omezení atributu	$\forall A.C$
Mapování hodnot atributů	$A = B, A \neq B$

Tabulka 3.3: Syntaxe konstruktorů

Konstruktor	Sémantika
Konjunkce	$C^I \cap D^I$
Disjunkce	$C^I \cup D^I$
Negace	$C^I \setminus D^I$
Existenční omezení	Množina instancí $x \in \Delta^I$, pro které platí $R^I(x) \cap C^I \neq \emptyset$
Univerzální omezení	Množina instancí $x \in \Delta^I$, pro které platí $R^I(x) \subseteq C^I$
Omezení počtu (kardinalita)	Množina instancí $x \in \Delta^I$, pro které platí $ R^I(x) \geq n$, nebo $ R^I(x) \leq n$
Existenční omezení atributu	Množina instancí $x \in \text{dom}(A^I)$, pro které platí $A^I(x) \in C^I$
Univerzální omezení atributu	Množina instancí $x \in \Delta^I$, pro které platí $x \in \text{dom}(A^I) \Rightarrow A^I(x) \in C^I$
Mapování hodnot atributů	Množina instancí $x \in \Delta^I$, pro které platí $A^I(x) = A^I$, nebo $A^I(x) \neq A^I$

Tabulka 3.4: Sémantika konstruktorů

Kapitola 4

RDF

RDF¹ (Resource Description Framework), volně přeloženo jako formát pro popis zdrojů, je standardem konsorcia W3C pro reprezentaci webových zdrojů. Podle oficiální definice se jedná o „obecný rámec pro popis, výměnu a znovupoužití metadat“.

Vznik tohoto formátu souvisí s dalším rozšiřováním sémantického webu (konec 90. let 20. století). Hlavní podmínkou tohoto rozšíření byla kompatibilita sémantických metadat se standardem pro metadata vytvořeným konsorciem W3C. V roce 1999 proto vzniká přímo na půdě W3C formát RDF, jenž je určen právě pro reprezentaci struktur webových metadat.

4.1 Syntaxe RDF

Formát RDF nemá, na rozdíl od jiných jazyků, přímo definovanou syntaxi. Jedná se o jakousi abstraktní syntaxi, jež předpokládá, že každý webový zdroj má určité vlastnosti, které lze popsat, a na základě nichž můžeme o zdrojích vytvářet jednoduchá tvrzení. Tato tvrzení mají formu trojic složených ze subjektu, predikátu a objektu. Citováno z [5].

- *Subjekt* představuje věc, kterou chceme pomocí RDF popsat. Může jí být jakýkoliv zdroj, kterému lze přiřadit URI a může tedy být identifikován pomocí URI reference.
- *Predikát* se někdy označuje jako vlastnost tvrzení a popisuje vztah mezi subjektem a objektem.
- *Objekt* potom představuje hodnotu dané vlastnosti tvrzení.

Způsobů, jak tuto abstraktní syntaxi reprezentovat, je několik. V následujícím textu si podrobněji přiblížíme některé z nich, včetně asi nejpoužívanějšího způsobu zápisu RDF trojic pomocí XML.

¹<http://www.w3.org/RDF>

4.1.1 N-Triples

První ze syntaxí, které si představíme, je N-Triples. Základem je řádkový formát, což znamená, že na každém řádku je zapsána jedna trojice ve formátu:

```
<subjekt> <predikát> <objekt>
```

V případě, že je objekt tvořen literálem, zapíše se tento namísto do závorek < > do uvozovek. K literáru je často nutné přiřadit i jeho typ. To učiníme pomocí dvojice znaků ^^. V níže uvedeném příkladě je zápis objektu s hodnotou číslo 7 a typem čísla integer:

```
<subjekt> <predikát> "7" ^^ xsd:integer
```

Hlavní nevýhodou tohoto způsobu zápisu je velký objem dat. URI reference identifikující subjekt a predikát mohou být dlouhé řetězce, což je příčinou i menší přehlednosti kódu. Výhodou je naopak snadné strojové zpracování nebo generování. Uvedený příklad cituje z [5].

```
<http://www.vip.cz/novak>
  <http://www.pojmy.cz/adresa> <_adresa>
<_adresa> <http://www.pojmy.cz/mesto> "Brno"
<_adresa> <http://www.pojmy.cz/ulice> "Česká 5"
<_adresa> <http://www.pojmy.cz/psc> "621 00"
```

4.1.2 Notation 3

Další možností je syntaxe Notation 3, zkráceně N3. Na první pohled se velmi podobá předešlé syntaxi N-Triples. Jedná se opět o zápis v podobě trojic subjekt – predikát – objekt.

```
<subjekt> <predikát> <objekt>
```

V syntaxi N3 lze používat prefixi jmenných prostorů s relativními URI referencemi (viz. kapitola 1.4.4) Z URI reference snadno vytvoříme klíčové slovo přidáním *@prefix*. Relativní URI reference potom vypadá v zápise následovně:

```
@prefix: URI reference
```

Je třeba zdůraznit, že jestliže je použit zkrácený zápis části trojice pomocí prefixu, tento se již neuzavírá do závorek < >. Za klíčové slovo se obecně považuje každý výraz uvezený znakem @ (zavináč).

Prázdné uzly mají obdobné označení jako relativní URI refence s tím rozdílem, že namísto prefixu se uvádí pouze znak `_` (podtržítko). Formát zápisu prázdného uzlu je tento:

```
_: název uzlu
```

K dalším rozdílům mezi syntaxí N3 a N-Triples patří možnost zapsat predikáty a objekty náležící k témuž jednomu subjektu tak, že v zápise uvedeme subjekt pouze jednou a následně pak seznam uspořádaných dvojic predikát – objekt, vzájemně oddělených středníkem. Stejně snadno lze zapsat i skupinu objektů náležících k jedné dvojici subjekt – predikát. Ovšem v tomto případě od sebe objekty oddělujeme pomocí čárek. Následující příklad cituje z [5].

```
@prefix vip: <http://www.vip.cz#>
    @prefix poj: <http://www.pojmy.cz#>
vip:novak poj:adresa _:adresa .
_adresa poj:mesto "Brno" ;
poj:ulice "Česká 5" ; poj:psc "621 00" .
```

4.1.3 RDF/XML syntaxe

Asi nejpoužívanější syntaxí je tzv. XML syntaxe RDF, často označována jako RDF/XML. Pro potřeby sémantického webu se zatím jeví jako nejvhodnější. K zápisu tvrzení RDF reprezentovaných RDF grafem (více viz. kapitola 4.2) se využívají elementy a jejich obsah, ale i atributy spolu s jejich hodnotami.

RDF graf lze graficky znázornit jako strom s konečnou množinou cest od kořene k listům. Tuto množinu je pak možné zapsat pomocí XML v podobě vnořených elementů. Hlavním elementem XML schematu je subjekt (hlavní uzel stromu). Prvním vnořeným elementem bude predikátová hrana tohoto subjektu. Její hodnota může být objektem tohoto tvrzení. Následujícím subjektem se stane objekt předešlého tvrzení. Stejným způsobem je možné pokračovat dále.

Přesný princip převodu trojic RDF grafu je tento, [5]:

1. Na prvním řádku je uvedeno povinné označení XML dokumentu a jeho verze.
2. Na dalším řádku začíná obsah elementu `<rdf:RDF>`, který říká, že následující obsah XML dokumentu se bude týkat RDF. Prvním z atributů tohoto elementu je deklarace jmenného prostoru `xmlns:rdf` označující, že všechny odkazy doplněné o prefix `rdf` jsou součástí jmenného prostoru definovaného URI referencí, která je hodnotou tohoto atributu. Následuje deklarace smyšleného jmenného prostoru `xmlns:poj`.
3. Další element `<rdf:Description>` je základní element, který označuje popis nějakého zdroje definovaného pomocí atributu `<rdf:about>`. Hodnota tohoto atributu představuje *subjekt* tvrzení.

4. Následující vnořený element `<poj:datum-vytvoreni>` představuje vlastnost (*predikát*) tvrzení a jeho hodnota „20.Března 2007“ představuje *objekt* tvrzení.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:poj="http://www.pojmy.cz/">

  <rdf:Description
    rdf:about="http://www.spolecnost.cz/cenik.html">
    <poj:datum-vytvoreni>20.Března 2007</poj:datum-vytvoreni>
  </rdf:Description>
</rdf:RDF>
```

4.2 RDF grafy

Základním stavebním kamenem RDF jsou tedy uspořádané trojice subjekt, predikát a objekt. Konečná množina těchto trojic definuje graf RDF. Tento je orientovaný graf složený z uzlů. Platí, [5]:

- Každá trojice je reprezentována dvojicí uzlů (nebo dvojicí uzel a list) a orientovanou spojnici (hranou).
- Hrana je vždy orientována směrem od subjektu k objektu a vyjadřuje vztah mezi subjektem a objektem.
- Objekty tvrzení mohou vystupovat jako subjekty jiného tvrzení nebo mohou být listem grafu.
- Každou trojici RDF můžeme spojit s jinou trojicí, aniž by se změnil její význam, bez ohledu na celkovou složitost grafu.



Obrázek 4.1: Trojice subjekt – predikát – objekt

Jednotlivé části RDF trojice mohou být v grafu reprezentovány pomocí URI reference. V neposlední řadě musíme uvést ještě jiné formy zápisu. Subjekt i objekt mohou být reprezentováni prázdným uzlem, objekt navíc i řetězem literálů.

4.2.1 Datové typy

Datové typy slouží v RDF pro reprezentaci hodnot jako jsou celá čísla, čísla s plovoucí desetinnou čárkou, apod. a dat. Datové typy se skládají ze tří částí. Z lexikálního prostoru, prostoru hodnot a lexikálně-hodnotového přiřazení.

Lexikální prostor datového typu je množina řetězců znaků Unicode. Lexikálně-hodnotové přiřazení datového typu je množina uspořádaných dvojic, kde první část uspořádané dvojice náleží do lexikálního prostoru datového typu a druhá část do prostoru hodnot datového typu. Rovněž platí, [5]:

- Každý prvek lexikálního prostoru může být přiřazen pouze k jednomu prvku z prostoru hodnot.
- Každý prvek z prostoru hodnot může být přiřazen k jakémukoliv počtu prvků (i žádnému) z lexikálního prostoru (lexikální reprezentace hodnoty).

Jako příklad si uvedeme datový typ XML Schematu *xsd:boolean*. V lexikálně-hodnotovém přiřazení datového typu má každý prvek prostoru hodnot dvě lexikální reprezentace.

- Prostor hodnot:
{T, F}
- Lexikální prostor:
{"0", "1", "true", "false"}
- Lexikálně-hodnotové přiřazení:
{("false", F), ("0", F), ("true", T), ("1", T)}

RDF nemá definován žádný vnitřní koncept pro reprezentaci čísel a jiných běžných hodnot. Využívá datové typy, které jsou definované odděleně a které jsou identifikovány pomocí URI reference. Běžně také využívá datové typy předdefinované pro XML Schéma. Ovšem nelze využívat všech datových typů, neboť některé nejsou pro používání v RDF vhodné.

Další věcí, která není v RDF nijak definována, je standardní postup pro definování nových datových typů. Díky XML Schématu je možné stávající datové typy rozšiřovat. V rámci RDF je definován pouze jeden jediný datový typ a to *rdf:XMLLiteral*. Ten se používá ke vkládání značek XML do RDF.

4.2.2 Literály

Literály slouží k identifikaci hodnot, tedy čísel a dat, prostřednictvím lexikální reprezentace. Vše, co je reprezentováno ve formě literálu, může být reprezentováno pomocí URI. V rámci tvrzení RDF může literál vystupovat pouze jako objekt. Literálů rozlišujeme dva typy, jednoduchý a typovaný literál.([5])

- *Jednoduchý literál* je řetězec znaků Unicode, ke kterému může být přidána značka konkrétního jazyka.

- *Typovaný literál* je také řetězec znaků Unicode, za kterým však musí být povinně uvedena URI reference datového typu. Označuje prvek identifikovaného prostoru hodnot datového typu získaného pomocí lexikálně-hodnotového přiřazení k řetězci literálu.

V následující tabulce jsou pro příklad uvedeny typové literály, jenž mohou být definovány pomocí datového typu *xsd:boolean* XML Schématu.

Typový literál	Lexikálně-hodnotové přiřazení	Hodnota
<xsd:boolean, "0">	<"0", F>	F
<xsd:boolean, "false">	<"false", F>	F
<xsd:boolean, "1">	<"1", T>	T
<xsd:boolean, "true">	<"true", T>	T

Tabulka 4.1: Typové literály

4.2.3 Prázdné uzly

Speciální variantou uzlu v RDF grafu je prázdný uzel. Od uzlů, jenž představují subjekty a objekty, se liší tím, že k těmto není přiřazena žádná URI reference. Vnitřní struktura RDF nijak neupřesňuje, jak tyto uzly označovat. Uzly mohou být zcela neoznačené, ale je vhodnější nějak tyto uzly označit, aby byly identifikovatelné. Příkladem využití takového typu uzlu je uzel v RDF grafu, zachycujícího informace o adrese osoby. Pojem „adresa“ není v takovém případě důležitý. Zajímá nás především ulice, město a poštovní směrovací číslo. Tyto jsou objekty subjektu „adresa“ a jejich hodnoty jsou právě středem našeho zájmu. Subjekt tvrzení o adrese bude v RDF grafu vyjádřena pomocí prázdného uzlu.

Kapitola 5

RDF Schéma

RDF Schéma¹ (zkráceně RDFS) není klasickým ontologickým jazykem. Jedná se o „sémantické rozšíření formátu RDF“ [5]. RDFS bylo vytvořeno v roce 1999 přímo konsorciem W3C. Jde o nadstavbu, která doplňuje do struktury RDF hlavní konstrukce z rámcových (či objektových) systémů. Jedná se o třídy a binární sloty s možností stanovit definiční obor a obor hodnot. Kromě toho může být nad třídami i sloty definována hierarchie. Zdroje z RDF pak přiřazujeme třídám z RDFS jako jejich instance (pomocí atributu *Type*).

RDFS je intuitivní, což vyhovuje především webovým návrhářům při zachycení sémantiky obsahu stránek. Nevýhodou oproti tradičním ontologickým jazykům je neschopnost RDFS precizně specifikovat podmínky příslušnosti ke třídám (lokální omezení) a zcela postrádá datové typy. Plná verze dále zahrnuje možnost reifikace tvrzení (tzn. transformace tvrzení jazyka na atomický objekt). Bohužel na druhé straně dochází k projevům nežádoucích vlastností z hlediska možností odvozování.

5.1 Třídy

K popisu vlastností zdrojů slouží v RDFS systém tříd (tab. 5.1) a jejich vlastností. V této kapitole se budeme zabývat pouze hlavními RDFS třídami. Kompletní seznam tříd a vlastností je k dispozici v [6].

Jednotlivé zdroje lze rozdělit do skupin, které označujeme jako třídy. Členy těchto tříd nazýváme instancemi. Rovněž platí, že i třída může být sama o sobě zdrojem identifikovaným pomocí URI reference, jež lze popsat RDF vlastností.

Množinu instancí každé třídy označujeme jako rozšíření třídy. Platí [5]:

- Dvě různé třídy mohou mít stejné množiny instancí, ale stále budou různými třídami.

Rozdíl mezi těmito třídami je v popisu pomocí různých vlastností. V dalším textu budeme pracovat s jmenným prostorem *wont* (wine ontology), definovaným URI referencí

¹<http://www.w3.org/TR/rdf-schema>

<http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine#>. Následující text a příklady vychází a citují z [5].

Jednotlivá RDF tvrzení zapisujeme pomocí jednoduchých příkazů. Fakt, že zdroj je instancí nějaké třídy zapíšeme pomocí RDF vlastnosti *rdf:type*. Skupina zdrojů, jenž jsou třídami RDFS, je třída označená pomocí *rdfs:Class*. Jestliže tedy chceme definovat RDFS třídu, sestavíme tvrzení o nějakém zdroji, který tuto třídu reprezentuje, a přiřadíme mu vlastnost *rdf:type*. Tato vlastnost má právě hodnotu zdroje *rdfs:Class*.

V našem případě definujeme zdroj *Wine* jako třídu RDFS:

```
wont:Wine rdf:type rdfs:Class
```

Třída	Popis
rdfs:Class	Třída všech zdrojů, které jsou RDFS třídami. Třída <i>rdfs:Class</i> je instancí sebe sama.
rdfs:Resource	Cokoliv popsaného pomocí RDF se nazývá zdrojem a je instancí této třídy. Tato třída obsahuje vše, jakákoliv další třída je podtřídou této třídy. Třída <i>rdfs:Resource</i> je instancí třídy <i>rdfs:Class</i> .
rdfs:Literal	Třída literálů, jejichž hodnoty jsou řetězce nebo čísla. Tato třída je instancí třídy <i>rdfs:Class</i> , ale je podtřídou třídy <i>rdfs:Resource</i> .
rdfs:Datatype	Třída datových typů. Je instancí i podtřídou třídy <i>rdfs:Class</i> . Ovšem každá instance této třídy je pouze podtřídou třídy <i>rdfs:Literal</i> .
rdf:XMLLiteral	Třída speciálních datových typů nazvaných XML literál. Tato třída je instancí třídy <i>rdfs:Datatype</i> a podtřídou třídy <i>rdfs:Literal</i> .
rdf:Property	Třída RDF vlastností. Je instancí třídy <i>rdfs:Class</i> .

Tabulka 5.1: RDFS třídy

5.1.1 Podtřídy

V rámci hierarchické struktury RDFS je možné definovat vedle tříd také tzv. podtřídy. Tato vlastnost, které označuje, že daná třída je podtřídou třídy jiné, je transitivní. Platí tedy, že pokud je třída T_1 podtřídou třídy T , potom všechny instance třídy T_1 jsou zároveň instancemi třídy T . Tento vztah zapisujeme pomocí *rdfs:subClassOf*.

V rámci naší cvičné ontologie je definována třída *RedWine* jako podtřída třídy *Wine*. Další třída, třída *DryRedWine*, je definována jako podtřída třídy *RedWine*. Z níže uvedeného zápisu

vyplývá, že suchá červená vína patří do skupiny vín červených. Zároveň také patří do třídy všech vín.

```
wont:RedWine rdfs:subClassOf wont:Wine
wont:DryRedWine rdfs:subClassOf wont:RedWine
```

5.2 Vlastnosti

RDF vlastnost je (ve specifikaci RDFS) relace mezi zdrojem subjektu a zdrojem objektu. Tyto vlastnosti jsou popsány pomocí třídy *rdf:Property* a některou z RDFS vlastností, *rdfs:range*, *rdfs:domain* nebo *rdfs:subPropertyOf*. Všechny vyjmenované vlastnosti jsou v RDF instancemi třídy *rdf:Property*. Novou vlastnost deklarujeme následujícím způsobem. Pomocí vlastnosti *rdf:type* přiřadíme zdroj ke třídě *rdf:Property*.

```
wont:hasMaker rdf:type rdf:Property
```

5.2.1 Vlastnost *rdfs:range*

Vlastnost *rdfs:range* říká, že hodnota (popřípadě hodnoty) deklarované vlastnosti bude zároveň instancí třídy, která vystupuje v tvrzení jako objekt, v němž je jako RDF vlastnost použita právě vlastnost deklarovaná pomocí *rdfs:range*.

V našem příkladě definujeme třídu *WineColor* a vlastnost *hasColor*. Poslední řádek zápisu říká, že pokud bude v tvrzení použita právě vlastnost *hasColor*, objekty v tomto tvrzení budou instancemi třídy *WineColor*.

```
wont:WineColor rdf:type rdfs:Class
wont:hasColor rdf:type rdf:Property
wont:hasColor rdfs:range wont:WineColor
```

5.2.2 Vlastnost *rdfs:domain*

Další z vlastností je *rdfs:domain*. Tato vlastnost se na rozdíl od vlastnosti předchozí vlastnosti vztahuje k subjektům. Každý zdroj, jemuž je přiřazena vlastnost pomocí *rdfs:domain*, je instancí jedné nebo více tříd.

Opět si uvedeme příklad. Nejprve definujeme třídu *Wine* a vlastnost *hasWineDescriptor*. Poslední tvrzení říká, že v tvrzení, v němž je použita vlastnost *hasWineDescriptor*, bude subjekt vždy instancí třídy *Wine*.

```
wont:Wine rdf:type rdfs:Class
wont:hasWineDescriptor rdf:type rdf:Property
wont:hasWineDescriptor rdfs:domain wont:Wine
```

5.2.3 Vlastnost `rdfs:subPropertyOf`

V RDFS lze definovat nejen vztahy mezi třídami, ale také vztahy mezi vlastnostmi. K tomu slouží vlastnost *subPropertyOf*. Pomocí této vlastnosti můžeme zapsat fakt, že jedna vlastnost je speciálním případem jiné, obecnější vlastnosti.

Nejprve si definujeme vlastnost *hasSugar* jako „pod-vlastnost“ vlastnosti *hasWineDescriptor*. Všechny instance třídy definované vlastností *hasSugar* budou zároveň i instancemi tříd definovaných vlastností *hasWineDescriptor*.

```
wont:hasWineDescriptor rdf:type rdf:Property
wont:hasSugar rdf:type rdf:Property
wont:hasWineDescriptor rdfs:subPropertyOf wont:hasSugar
```

Kapitola 6

OWL

Na základě zkušeností s vývojem a využíváním DAML+OIL vznikl pod hlavičkou W3C Ontology Working Group nový projekt, který položil základy novému ontologickému jazyku OWL¹ (Ontology Web Language). První verze byla zveřejněna v květnu roku 2002. Od té doby byl jazyk několikrát upravován. V současnosti dochází k postupnému ustálení jeho podoby.

6.1 Verze jazyka OWL

Jazyk OWL je dostupný v několika verzích, které se od sebe vzájemně liší v závislosti na uživatelských skupinách s různou úrovní schopností nebo s různými nároky na implementaci konkrétního projektu.

První verzí je OWL Lite. Tato verze byla navržena pro jednoduchou implementaci. Lze vytvořit základní klasifikační hierarchickou strukturu dokumentu s jednoduchými omezeními. OWL Lite využívá jen některé prvky jazyka OWL a obecně se řídí omezeními definovanými pro verzi OWL DL. Je vhodná i pro začínající uživatele.

Druhou verzí je OWL DL (DL - deskripční logika). OWL DL je zvláště vhodná pro rozhodovací systémy. Poslední z verzí je OWL Full. Jedná se o kompletní jazyk OWL. Platí, že [5]:

- Každá platná OWL Lite ontologie je platnou OWL DL ontologií.
- Každá platná OWL DL ontologie je platnou OWL Full ontologií.

Verze OWL Full a OWL DL podporují stejné konstruktory jazyka OWL. Rozdíly mezi nimi spočívají v omezení možností použití některých z těchto prvků a dále v používání prvků RDF.

OWL Full je, jak již bylo řečeno, kompletní verzí jazyka OWL bez jakýchkoli omezení. Je určen pro ty uživatele, kteří očekávají maximální výrazovou schopnost jazyka a volnost

¹<http://www.w3.org/2004/OWL>

při tvorbě ontologií. Dovoluje zcela libovolně propojovat prvky OWL s prvky RDFS. Oddělování tříd, vlastností, instancí a datových typů není, podobně jako je tomu u RDFS, tak striktní. Je navržena tak, aby kompatibilita s formátem RDF byla maximální.

U OWL DL dochází k jakémusi omezení propojování prvků OWL a RDF. Rovněž je nutné, aby třídy, vlastnosti, instance a datové typy byly vzájemně odděleny. Hlavním důvodem, proč vznikla tato verze jazyka OWL, je existence mocných rozhodovacích systémů, jenž podporují ontologie omezené pravidly platnými právě v OWL DL.

6.2 Syntaxe

OWL je tvořena pomocí RDF trojic a lze tedy vytvářet RDF grafy. RDF trojice jsou reprezentovány RDF/XML syntaxí (jakoukoli formou této syntaxe), jež je doporučena konsorciem W3C.

Na příkladě třídy *Wine* v OWL si ukážeme, jak taková reprezentace pomocí RDF/XML může vypadat.

```
<owl:Class rdf:ID="Wine"/>
```

Pokud bychom použili jiný typ syntaxe RDF/XML, získali bychom zápis stejného tvzení například v této podobě:

```
<rdf:Description rdf:about="Wine">  
  <rdf:type rdf:Resource="http://www.w3.org/2002/07/owl#Class"/>  
</rdf:Description>
```

6.3 Struktura OWL dokumentu

Struktura OWL dokumentu zachycuje jednotlivé pojmy ontologie. Dokument obsahuje vždy pouze jednu hlavičku. Dále následují definice tříd, vlastností a i instancí (individuí).

6.3.1 Hlavička

Hlavička obsahuje důležité informace o ontologii, jíž se dokument týká. Dále může obsahovat komentáře, informace o verzi dokumentu či o vnořených ontologiích.

Následuje ukázka hlavičky OWL dokumentu patřícího výše zmíněné ontologii o vínech.

```
<owl:Ontology rdf:about="">  
  <rdfs:comment>An example OWL ontology  
</rdfs:comment>
```

```

<owl:priorVersion>
  <owl:Ontology rdf:about=
    "http://www.w3.org/TR/2003/WD-owl-guide-20030331/wine"/>
</owl:priorVersion>
<owl:imports rdf:resource=
  "http://www.w3.org/TR/2003/CR-owl-guide-20030818/food"/>

<rdfs:comment>
  Derived from the DAML Wine ontology at
  http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml
  Substantially changed, in particular the Region based relations.
</rdfs:comment>
<rdfs:label>Wine Ontology</rdfs:label>
</owl:Ontology>

```

- Hlavička je uzavřena do značek `<owl:Ontology>`. Atribut `rdf:about` je prázdný atribut. Vyjadřuje, že popisovaný subjektem bude vlastní dokument.
- Značka `<rdfs:comment>` označuje komentář. Jeho obsah nijak neovlivňuje strukturu dokumentu.
- Značka `<owl:priorVersion>` označuje předchozí verzi dokumentu. Dále může hlavička obsahovat ještě značku `<owl:versionInfo>`. Ta obsahuje označení aktuální verze.
- Značka `<owl:imports>` se využívá k importování jiné ontologie. Atributem této značky je `rdf:resource`.
- Značka `<rdfs:label>` má obdobný význam jako `rdfs:comment`. Slouží k popisu zdroje způsobem čitelným pro člověka.

6.3.2 Třídy

Základním hierarchickým prvkem každé ontologie je třída. Třída je uspořádaná struktura (množina) prvků, jenž spolu nějak souvisejí. Tyto prvky nazýváme instancemi (individu) třídy. Každou třídu jednoznačně popisuje její název (URI) a soubor prvků, které k ní náleží. Třidu lze definovat několika způsoby, [5]:

- Identifikátorem třídy (URI reference).
- Výčtem prvků, které budou tvořit instance třídy.
- Omezením vlastností.
- Sjednocením nebo průnikem dvou a více definicí tříd.
- Doplnkem definice třídy.

V následujícím textu (vycházíme z textu [5]) ukážeme, jak se od sebe jednotlivé způsoby liší.

Třídy definované identifikátorem

Třída je definována pouze svým jménem. Neobsahuje žádné prvky ani vlastnosti. V zápise vypadá definice takto:

```
<owl:Class rdf:ID="Wine"/>
```

Třídy definované výčtem prvků

Třída definovaná tímto způsobem je třída anonymní. Na tyto třídy nelze přímo odkazovat. Uplatnění najdou ve složitějších konstrukcích (sjednocení, průnik, atd.).

V ukázce je pomocí výčtu prvků *White*, *Rose* a *Red* definována třída *WineColor*. Vlastnost *owl:oneOf* obsahu seznam individuů, které jsou instancemi definované třídy.

```
<owl:Class rdf:ID="WineColor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Wine"/>
    <owl:Thing rdf:about="#Rose"/>
    <owl:Thing rdf:about="#Red"/>
  </owl:oneOf>
</owl:Class#>
```

Třídy definované omezením vlastností

Speciální definice třídy pomocí omezení vlastností, specifikuje anonymní třídu všech individuů, jež budou odpovídat podmínkám omezení. Omezení existují dva druhy, omezení hodnoty a omezení kardinality. Obecný formát definice třídy tímto způsobem vypadá takto:

```
<owl:Restriction>
  <owl:onProperty rdf:resource="vlastnost"/>
  (řádek s omezením vlastnosti nebo kardinality)
</owl:Restriction>
```

Omezení hodnotou využívá vždy některou ze tří základní owl vlastností. Tyto jsou uvedeny v následující tabulce, [5].

OWL vlastnost	Popis
owl:allValuesFrom	Popisuje třídu všech individuí, pro která daná vlastnost nabývá pouze hodnot definovaných omezením (může to být i třída).
owl:someValuesFrom	Popisuje třídu všech individuí, pro která alespoň jedna hodnota dané vlastnosti nabývá hodnoty definované omezením.
owl:hasValue	Popisuje třídu všech individuí, pro která daná vlastnost nabývá hodnoty definované omezením (pouze jedna konkrétní hodnota).

Tabulka 6.1: Omezení hodnotou

Vše si ukážeme na příkladu třídy definované omezením vlastnosti *hasSugar*, jež musí nabývat hodnoty *Sweet*.

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasSugar"/>
  <owl:hasValue rdf:resource="#Sweet"/>
</owl:Restriction>
```

Stejně je tomu tak i tříd definovaných omezením kardinality. Opět jsou k dispozici tři vlastnosti, které lze využít při omezování kardinality. Jsou uvedeny v následující tabulce, [5].

OWL vlastnost	Popis
owl:maxCardinality	Popisuje třídu všech individuí, pro která daná vlastnost nabývá maximálně počtu hodnot definovaných omezením.
owl:minCardinality	Popisuje třídu všech individuí, pro která daná vlastnost nabývá alespoň počtu hodnot definovaných omezením.
owl:Cardinality	Popisuje třídu všech individuí, pro která daná vlastnost nabývá přesně počtu hodnot definovaných omezením.

Tabulka 6.2: Omezení kardinality

V příkladě je uvedena třída definovaná omezením vlastnosti *madeFromGrape* na kardinalitu 1. Znamená to, že víno je vyráběno pouze z jednoho typu hroznů.

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#madeFromGrape"/>
  <owl:maxCardinality rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1
  </owl:maxCardinality>
</owl:Restriction>
```

Třídy definované průnikem

Dále lze definovat třídy pomocí množinových operací nad třídami. K definici je použito vlastnosti *owl:intersectionOf*.

Třída *White Wine* je průnik dvou tříd. První je třída *Wine*, jejímiž instancemi jsou všechna vína. Druhou je třída *hasColor* s hodnotou omezenou na hodnotu *White*, tedy na všechna bílá vína.

```
<owl:Class rdf:ID="WhiteWine">
  <owl:intersectionOf rdf:parseType="Collection"/>
    <owl:Class rdf:about="#Wine"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasColor"/>
      <owl:hasValue rdf:resource="#White"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Třídy definované sjednocením

Třída, jež je definována sjednocením, obsahuje prvky, které náleží alespoň jedné ze sjednocených tříd. Použitou vlastností v tomto případě je *owl:unionOf*.

Níže uvedená definice ukazuje třídu *WineDescriptor*. Ta vznikla sjednocením prvků tříd *WineTaste* a *WineColor*.

```
<owl:Class rdf:ID="WineDescriptor">
  <owl:unionOf rdf:parseType="Collection"/>
    <owl:Class rdf:about="#WineTaste"/>
    <owl:Class rdf:about="#WineColor"/>
  </owl:unionOf>
</owl:Class>
```

Třídy definované doplňkem

Posledním z uvedených typů možných definic je definice třídy pomocí doplňku. Od předchozích definic se liší tím, že lze použít pouze jednu třídu, k níž budeme vytvářet doplněk. Jako výsledek můžeme získat velmi velkou třídu obsahující značné množství prvků.

Bohužel v našem použitém příkladě žádná, tímto způsobem definovaná třída není k dispozici. Pravděpodobně by ani neměla význam. Pro cvičné účely vytvoříme „abstraktní“ třídu

ke třídě *WhiteWine*. Tato třída bude obsahovat všechna ostatní vína, tedy *Rose* a *RedWine*. V definici využijeme vlastnosti *owl:complementOf*.

```
<owl:Class>
  <owl:complementOf>
    <owl:Class rdf:about="#WhiteWine"/>
  </owl:complementOf>
</owl:Class>
```

Kromě těchto výše uvedených možností definic tříd existují ještě další možnosti, jak tyto definice upravit. Může využít těchto vlastností [5]:

- *rdfs:subClassOf* – označuje, že prvky definované třídy jsou podmnožinou prvků jiné třídy (nadřazené).
- *owl:equivalentClass* – označuje, že množina prvků jedné třídy je přesně stejná jako množina prvků jiné třídy.
- *owl:disjointWith* – označuje, že množina prvků jedné třídy neobsahuje žádné společné prvky s množinou prvků jiné třídy.

6.4 Vlastnosti

Vlastnosti jsou svým způsobem binární relace, díky nimž můžeme ke třídám i k jejich prvkům přidávat tvrzení a spojení mezi jednotlivými objekty a datovými typy. V zásadě rozlišuje dva typy vlastností. Vlastnosti objektové a dato-typové.

Označení objektová vlastnost patří vlastnosti, která spojuje dva objekty. Definujeme ji elementem *owl:ObjectProperty*. Dato-typová vlastnost je vlastnost, jež spojuje objekt s nějakým datovým typem. Je definována pomocí elementu *owl:DatatypeProperty*. Obecná definice má následující formát:

```
<owl:ObjectProperty rdf:ID="název vlastnosti">
  Další podmínky a typy vlastností
</owl:ObjectProperty>
```

Pokud chceme v OWL definovat vlastnosti, jsou nám k dispozici tyto konstruktory, [5]:

- RDFS konstruktory:
 - *rdfs:subPropertyOf*
 - *rdfs:domain*
 - *rdfs:range*

- Vztahy mezi vlastnostmi:
 - *owl:equivalentProperty*
 - *owl:inverseOf*
- Globální omezení kardinality:
 - *owl:FunctionalProperty*
 - *owl:InverseFunctionalProperty*
- Logické charakteristiky:
 - *owl:SymmetricProperty*
 - *owl:TransitiveProperty*

RDFS konstruktory byly popsány v předchozí kapitole 4.1.2, proto se jimi již nebudeme opět zabývat. Zaměříme se naopak na zbylé typy konstruktorů.

Vztahy mezi vlastnostmi

Prvním z konstruktorů této skupiny je *owl:equivalentProperty*. Slouží k označení vlastností se stejnými prvky (hodnotami). Neznamená to ovšem, že se jedná o stejné vlastnosti. Mohou se od sebe lišit svým významem.

Druhý konstruktor *owl:inverseOf* označuje vztahy v obou směrech, tedy doplnění jedné relace opačným případem. Pro upřesnění uvedeme příklad objektové vlastnosti *producesWine* a její inverzní vlastnosti *hasMakerf*.

```
<owl:ObjectProperty rdf:ID="producesWine">  
  <owl:inverseOf rdf:resource="#hasMaker"/>  
</owl:ObjectProperty>
```

Globální omezení kardinality

Konstruktor *owl:FunctionalProperty* definuje funkcionální vlastnost, která může nabývat pouze jedné jedinečné hodnoty pro každou instanci. K vymezení této hodnoty se často používá některý z konstruktorů *rdfs:domain* nebo *rdfs:range*.

Za příklad nám může posloužit objektová vlastnost *hasColor*. Pokud ji označíme funkcionální vlastností, říkáme, že její hodnotou může být jen jeden prvek ze třídy *WineColor*. Víno má tedy vždy jednu barvu.

```
<owl:ObjectProperty rdf:ID="hasColor">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineColor"/>
</owl:ObjectProperty>

<owl:FunctionalProperty rdf:about="hasColor"/>
```

K předešlému konstruktoru existuje inverzní funkcionální vlastnost *owl:InverseFunctionalProperty* omezující vlastnost právě v opačném směru. Neříká jakému objektu může vlastnost patřit, ale vymezuje hodnotu subjektu.

Logické charakteristiky

Symetrická vlastnost, definovaná konstruktorem *owl:SymmetricProperty*, označuje fakt, že pokud dvojice prvků (a, b) je instancí dané vlastnosti, pak také dvojice (b, a) je instancí téže vlastnosti. V praxi to znamená, že konstruktory *rdfs:domain* a *rdfs:range* obvykle vymezují stejnou hodnotu vlastnosti.

Druhou vlastností je transitivita. Je definována konstruktorem *owl:TransitiveProperty*. Tato vlastnost říká, že pokud dvojice prvků (a, b) je instancí dané vlastnosti a rovněž i dvojice (b, c) , pak platí, že dvojice (a, c) bude opět instancí téže vlastnosti.

Kapitola 7

Editory ontologií

Reakcí na vzrůstající zájem o ontologie a jejich implementaci byl vznik softwarových nástrojů umožňujících tvorbu ontologií i jejich následnou editaci. Pod softwarovými nástroji zde rozumíme jednak „klasické“ editory a jednak komplexní řešení, která jsou vhodná především při začlenění ontologií do databázově orientovaných podnikových informačních systémů. Během rozvoje ontologického inženýrství bylo vytvořeno již nesčetné množství různých podpůrných nástrojů. Ovšem u většiny z nich došlo krátce po jejich vzniku k zastavení dalšího vývoje. Jedním z důvodů byla například nejednotnost v názorech na jazyk, který by byl nejvhodnější pro reprezentaci znalostních informací. Poslední vývoj v této oblasti ukazuje, že favoritem se mezi jazyky stal jazyk OWL.

Jedním z editorů, které právě tento jazyk používají, je editor Protégé (více v kapitole 7.). V současnosti se jedná o nejpoužívanější editor. Jeho nespornou výhodou je, že je stále vyvíjen a doplňován o další přídavné balíčky.

V následující kapitole se podrobněji seznámíme s několika dalšími editory ontologií. Jedná se samozřejmě pouze o reprezentativní výběr, který je zaměřen výhradně na open-source software.

Název	Vytvořen URL
CMapTools Ontology Editor	University of Florida, IHMC http://cmap.ihmc.us/coe
DERI Ontology Management Environment	Ontology Management Working Group http://dome.sourceforge.net
Chimaera	Stanford University Knowledge Systems Laboratory http://www.ksl.stanford.edu/software/chimaera
Java Ontology Editor	University of South Carolina, Department of Computer Science and Engineering http://www.cse.sc.edu/research/cit/demos/java/joe/
Karlsruhe ontology	Universität Karlsruhe http://kaon.semanticweb.org
Karlsruhe ontology 2	Universität Karlsruhe, University of Manchester http://kaon2.semanticweb.org
Knoodl	Revelytix http://www.knoodl.com/ui/home.html
Ontolingua Ontology Editor	Stanford University http://www.ksl.stanford.edu/software/ontolingua
Protégé	Stanford University, University of Manchester http://protege.stanford.edu
Synaptica	Dow Jones http://www.synaptica.com/djcs/synaptica
Swoop	University of Maryland http://www.mindswap.org/2004/SWOOP
TopBraid Composer	TopQuadrant http://www.topquadrant.com/topbraid/composer/index.html

Tabulka 7.1: Přehled editorů ontologií

7.1 CMapTools Ontology Editor

CMapTools Ontology Editor (dále jen COE) je projekt probíhající na institutu IHMC University of Florida¹, jehož cílem je vytvoření sady nástrojů pro návrh, editaci a prohlížení RDF/OWL ontologií. COE je založen na IHMC CMapTools², což je nástroj pro práci se znalostními modely.

COE je volně šiřitelný software implementovaný v jazyce Java, který lze používat pod operačními systémy Windows, Apple a Sun. Poskytuje základní nástroje pro práci s ontologiemi, jako je editace a návrh nových ontologií, sdílení ontologií s dalšími uživateli v reálném čase, export do různých formátů (např. RDF/XML, N3 a další), ukládání grafů ve formě obrázků atd.

7.2 DERI Ontology Management Environment

DERI Ontology Management Environment (zkráceně DOME) je produkt vyvinutý skupinou Ontology Management Working Group³ (OMWG). Jedná se open-source software založený na platformě Eclipse⁴. Pracuje pod operačními systémy Windows a Linux.

Hlavním cílem bylo vytvoření nástroje, který by byl schopen účinně a efektivně spravovat ontologie a který by poskytoval řešení základních problémů. DOME obsahuje nástroje pro editaci, prohledávání, vývoj, slučování a grafickou prezentaci ontologií. DOME je možné dále rozšířit o plug-iny.

7.3 Chimaera

Chimaera je přes web dostupná aplikace vytvořená Stanford University Knowledge Systems Laboratory⁵. Je založena na Ontolingua Distributed Collaborative Ontology Environment⁶.

Chimaera je software umožňující návrh a následnou údržbu ontologií prezentovaných na Internetu. Ke spojení se serverem využívá protokolu OKBC⁷. Mezi hlavní funkce patří spojování různých ontologií. Dále lze importovat ontologie různých formátů, měnit strukturu ontologií, řešit konflikty jmen tříd a dalších částí ontologií, editovat ontologie a v neposlední řadě exportovat ontologie do formátů DAML nebo OWL.

¹<http://www.ihmc.us/index.php>

²<http://cmap.ihmc.us/conceptmap.html>

³<http://www.omwg.org>

⁴<http://www.eclipse.org>

⁵<http://www.ksl.stanford.edu>

⁶<http://www.ksl.stanford.edu/software/ontolingua>

⁷<http://www.ai.sri.com/okbc>

7.4 Java Ontology Editor

Java Ontology Editor (dále JOE) je ontologický editor distribuovaný formou java appletu. Tento editor, vytvořený na půdě University of South Carolina⁸, není již v současné době dále vyvíjen.

Editor JOE je určen pro ontologie uložené ve formátu KIF⁹. Hlavním cílem je poskytovat rozhraní pro grafickou reprezentaci ontologií v různých formátech. Prvním formát je velmi podobný struktuře Průzkumníka operačního systému Windows. Druhou možností je zobrazení ontologie ve formě stromové struktury. Výstupem třetího formátu je E-R diagram (Entity-Relationship diagram).

7.5 Karlsruhe ontology

Karlsruhe ontology, zkráceně KAON (často označován jako KAON1), je ontologický systém vytvořený na Universität Karlsruhe¹⁰ v roce 2002. Jde o open-source systém založený na platformě Java a zaměřený především na business aplikace.

KAON obsahuje komplexní sadu nástrojů pro snadné vytváření a správu RDF ontologií. Důležitou vlastností KAONu je škálovatelnost a možnost efektivního odvozování ontologií. Celá struktura editoru KAON lze rozdělit do tří modulů:

- *Frontend*. Tuto část reprezentují dvě aplikace, OI-modeler a KAON Portal. OI-modeler je editor pro jednoduchou práci s ontologickými modely. KAON Portal je portál určený pro uživatele. Obě dvě aplikace jsou volně šiřitelné a lze je snadno nakonfigurovat podle konkrétních požadavků uživatele.
- *Core of KAON*. Jádrem KAONu jsou dvě API rozhraní. První je určeno pro Resource Description Framework (RDF) a druhé pro KAON Ontology Language.
- *Libraries* zajišťují funkci KAONu. Mohou být používány samostatně i mimo kontext ontologicky orientovaných aplikací.

7.6 Karlsruhe ontology 2

Karlsruhe ontology 2, neboli KAON2, je nástupcem projektu KAON. Narozdíl od KAONu se jedná o systém pro správu OWL-DL, SWRL a F-Logic ontologií. Jedná se tedy o zcela nový systém a není tudíž kompatibilní s předchozím systémem KAON. KAON2 je již komerční, ale pro akademické účely je poskytován zdarma.

⁸<http://www.cse.sc.edu>

⁹<http://en.wikipedia.org/wiki/KIF>

¹⁰<http://www.uni-karlsruhe.de>

KAON2 nabízí například následující funkce:

- API pro práci s OWL-DL, SWRL a F-Logic ontologiemi.
- Modul pro extrakci ontologických instancí z relačním databází.
- DIG¹¹ rozhraní pro přístup k nástrojům jako je například Protégé.
- Nezávislý sever poskytující přístup k ontologiím pomocí RMI¹².

7.7 Knoodl

Knoodl je webová aplikace umožňující tvorbu, sdílení a editaci ontologií a RDF znalostních databází. Hlavním cílem této aplikace je spolupráce. K dispozici je tudíž rozsáhlá dokumentace ontologií ve formě „wikitextu“. Na jednotlivých ontologiích je možné spolupracovat s více lidmi v reálném čase. Ontologie, zde vytvořené lze rovněž stahovat a používat v jiných projektech. Knoodl podporuje jazyky OWL, RDF, RDFS a SPARQL¹³.

7.8 Ontolingua Ontology Editor

Ontolingua Ontology Editor je další přes web dostupná aplikace vytvořená Stanford University Knowledge Systems Laboratory. Ke spojení se serverem opět využívá protokolu OKBC.

Editor poskytuje prostředí, ve kterém je možné prohlížet, vytvářet, editovat a používat ontologie. Ontolingua Ontology Editor má širokou členskou základnu. V současné době přes 150 aktivních členů, které je možné požádat o pomoc při řešení problémů nebo o informace o jejich projektech.

7.9 Synaptica

Synaptika je další z webových aplikací. Hlavním důvodem vytvoření tohoto software bylo zjednodušení a standardizace „slovníku“ užívaného v managementu a obchodu. Editor podporuje OWL a SKOS¹⁴.

Synaptika poskytuje nástroje pro efektivní vytváření a údržbu podnikových taxonimií, tezaurů, jmenných katalogů atd. Je rovněž možné vytvářet nová metadata, filtry a různá pravidla. Výsledkem je sjednocení používaných pojmů a postupů a následně i zlepšení a zrychlení spolupráce se zákazníky.

¹¹<http://dig.sourceforge.net>

¹²<http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>

¹³<http://www.w3.org/TR/rdf-sparql-query>

¹⁴<http://www.w3.org/2004/02/skos>

7.10 Swoop

Swoop je prohlížeč a editor OWL ontologií vytvořený na půdě MIND Lab¹⁵ na University of Maryland. Jedná se o volně šiřitelný software implementovaný v jazyce Java.

Swoop je nástroj pro prohlížení, vytváření a editaci ontologií. Dále obsahuje sadu nástrojů pro vyhledávání a opravu chyb a nesrovnalostí v ontologiích. Swoop umožňuje vizualizaci ontologií ve formě grafů a export například do formátů RDF/XML a N3. Průběžně je doplňován o nejrůznější rozšíření ve formě plug-inů.

7.11 TopBraid Composer

TopBraid Composer je volně šiřitelný editor založený na platformě Eclipse. Jedná se o jeden z programů TopBraid SuiteTM¹⁶, což je soubor produktů pro sémantická řešení splňující standardy W3C.

TopBraid Composer je prostředím pro vytváření ontologií a vývoj sémantických aplikací. Plně podporuje RDFS, OWL, SWRL a SPARQL. K dispozici jsou dvě verze, Standard Edition, určený pro tvorbu a editaci ontologií, a Maestro Edition, určený pro tvorbu a správu webových aplikací založených na platformě TopBraid Live.

¹⁵<http://mindlab.umd.edu/home.php>

¹⁶<http://www.topquadrant.com/topbraid/index.html>

Kapitola 8

Editor Protégé

8.1 Instalace

Editor Protégé spadá do skupiny tzv. freeware softwarů. Je distribuován pod licencí open-source Mozilla Public License¹. Aktuální verze programu, v současné době Protégé 3.3.1, Protégé 3.4 beta a Protégé 4.0 beta, jsou volně ke stažení z internetových stránek <http://protege.stanford.edu/download/registered.html>.

Editor Protégé lze instalovat pod operační systémy Windows, Linux, MacOSX, Solaris a další. Nalezneme zde i verze instalačních balíčků s možností současné instalace Java Virtual Machine, který je nutný pro samotnou činnost programu. Na http://protegewiki.stanford.edu/index.php/Protege_Plugin_Library jsou také k dispozici přídavné plug-iny, jenž umožňují používání nadstandardních funkcí.

Užitečnými podpůrnými nástroji jsou například balíčky OWLViz a GraphViz. OWLViz je vizualizační balíček, díky němuž můžeme zobrazit graf hierarchie tříd. Balíček GraphViz nám pak slouží pro export těchto grafů do různých grafických formátů.



Obrázek 8.1: Logo Protégé

¹<http://www.mozilla.org/MPL>

8.2 The Protégé-Frames Editor

The Protégé-Frames Editor poskytuje kompletní uživatelské rozhraní a znalostní server, který slouží jako podpora uživatelům a na kterém jsou k dispozici trénovací data. The Protégé-Frames Editor implementuje znalostní modely kompatibilní s Open Knowledge Base Connectivity (OKBC). V takovém modelu se ontologie skládají ze sady tříd organizovaných v subsumpční hierarchie, sady slotů přidružených ke třídám a reprezentujícím jejich vlastnosti a vzájemné vztahy, a konečně ze sady instancí těchto tříd.

The Protégé-Frames Editor je navržen tak, aby co nejvíce vyhovoval potřebám uživatele. Uživatel si může přizpůsobit vzhled rozhraní. Dále nesmíme zapomenout na množství plug-inů a dalších podpůrných nástrojů, pomocí nichž je možné rozšířit popřípadě i doplnit funkce editoru. V neposlední řadě je k dispozici Java-based Application Programming Interface (dále jen API). API umožňuje jiným plug-inům a aplikacím přistupovat, využívat a zobrazovat ontologie vytvořené pomocí tohoto editoru.

K editoru náleží přirozeně dokumentace, jež je k nalezení na <http://protege.stanford.edu/doc/users.html>. Ta obsahuje i velmi populární Getting Started with Protégé-Frames a uživatelskou příručku Protégé-Frames User's Guide.

8.3 The Protégé-OWL Editor

The Protégé-OWL Editor je v podstatě rozšíření editoru Protégé o podporu Web Ontology Language (OWL). Tato verze editoru je reakcí na vzrůstající oblíbenost tohoto jazyka. Ten je stále rozvíjen a navíc je i schválen konsorciem W3C.

The Protégé-OWL Editor umožňuje přidávat a ukládat OWL a RDF ontologie. Dále je možné editovat a zobrazovat třídy, vlastnosti a SWRL² (Semantic Web Rule Language) pravidla.

Editor má velmi flexibilní architekturu. Je poměrně snadné si jej nastavit podle svých představ nebo jej rozšířit o další nástroje. Tento editor je těsně spjat s frameworkem Jena³ a má k dispozici Java API.

Ani v tomto případě nechybí dokumentace (<http://protege.stanford.edu/doc/users.html>).

8.4 Protégé Users

Editor Protége je uživatelsky velmi vstřícný. Hlavní předností jsou průběžně aktualizované domovské stránky editoru <http://protege.stanford.edu>, na kterých jsou k dispozici veškeré důležité informace. Uživatelé je k dispozici dokumentace zaměřené jak na běžného uživatele, tak na vývojáře. Na stránkách nalezneme záznamy z konferencí, seznam projektů užívajících editoru Protégé, seznam již vytvořených ontologií a seznam plug-inů. Uživatel má dále

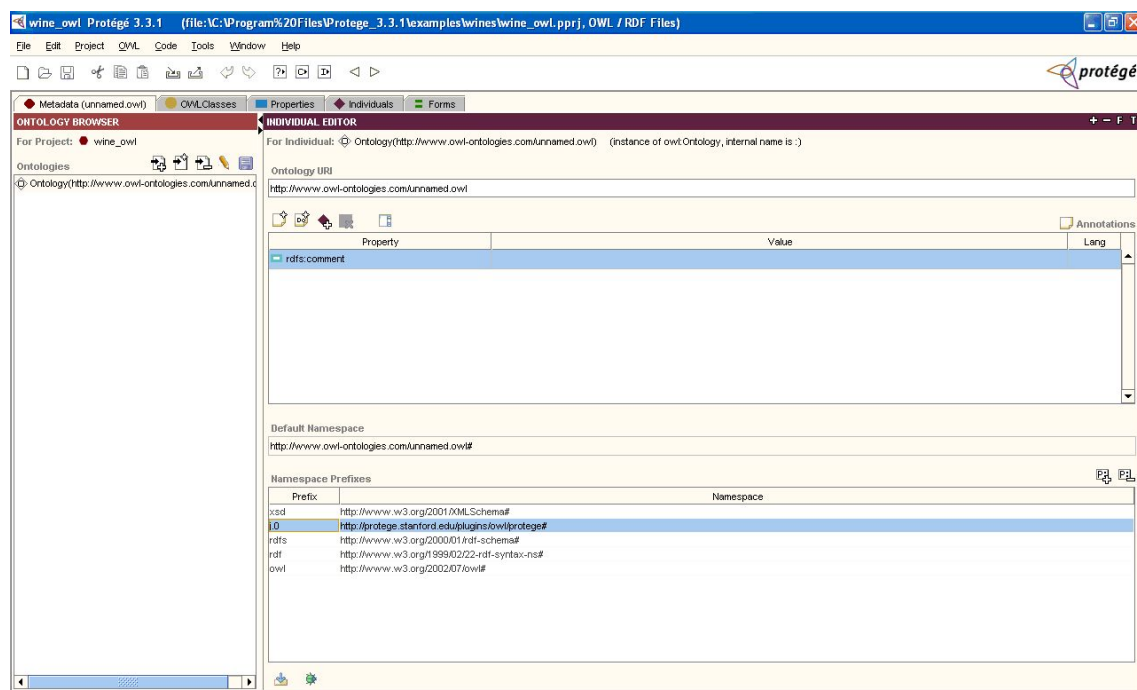
²<http://www.w3.org/Submission/SWRL>

³<http://jena.sourceforge.net>

možnost se s případnými problémy a dotazy obrátit na diskusní skupiny. V neposlední řadě musíme ještě zmínit Protege Wiki, http://protegewiki.stanford.edu/index.php/Main_Page. Ta v současné době přechází ze staré verze na novou, označovanou jako New Protégé Semantic MediaWiki. Cílem je zlepšení samotné struktury stránek a usnadnění práce s nimi.

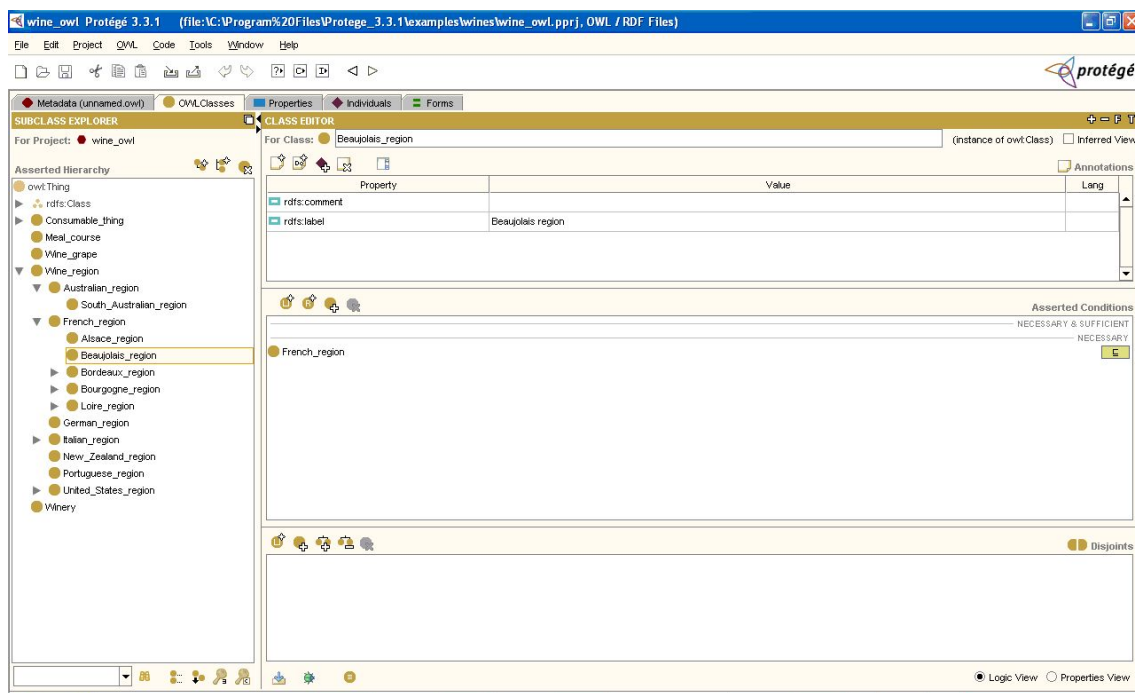
8.5 Uživatelské rozhraní

Editor Protégé má grafické uživatelské rozhraní. K dispozici jsou rozbalovací menu obsahující základní funkce, ikony zastupující nejpoužívanější funkce a záložky, které se týkají speciálních oblastí tvorby ontologie. K základním záložkám patří karty OWLClasses, Properties, Individuals, Forms pro práci s jednotlivými částmi ontologie a Metadata obsahující základní informace o ontologii.



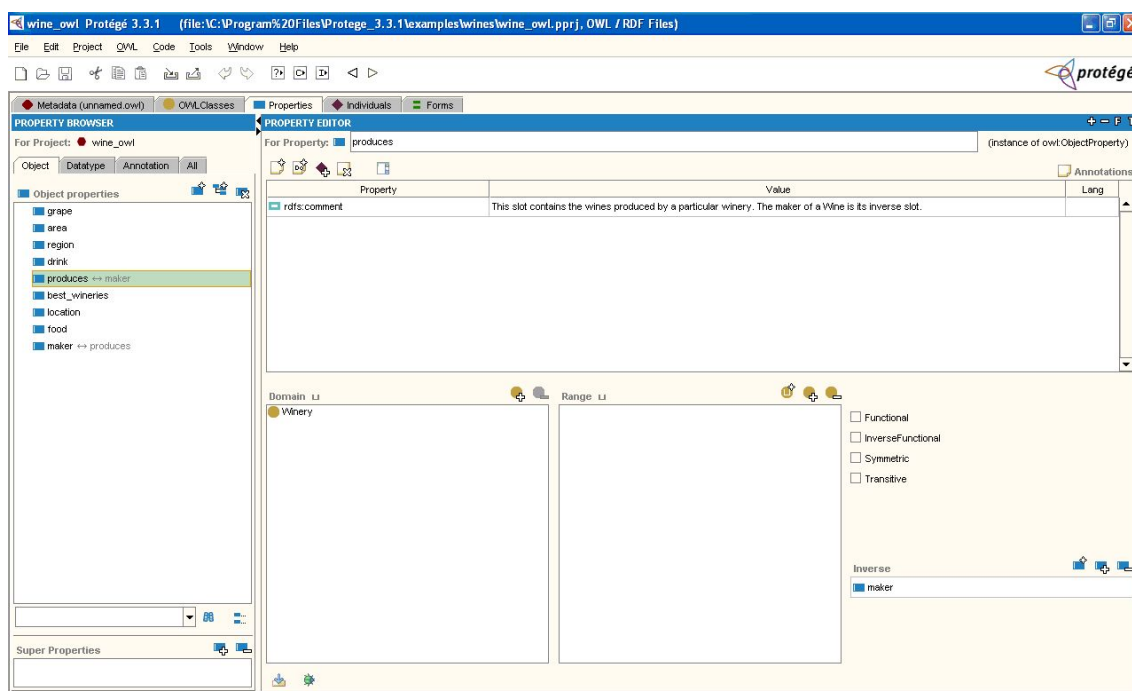
Obrázek 8.2: Karta Metadata

Záložka OWLClasses je určena pro práci s třídami. Karta je rozdělena na dvě části. V levé je umístěn uspořádaný strom tříd. Jako kořen je vždy vložena třída *owl:Thing*. Další přidané třídy jsou pak jejími podtřídami (subclasses). V druhé části se nachází editor, kde lze přidávat a editovat popisky třídy (*rdfs:comment*) a vkládat výrazy nebo omezení definující třídu.



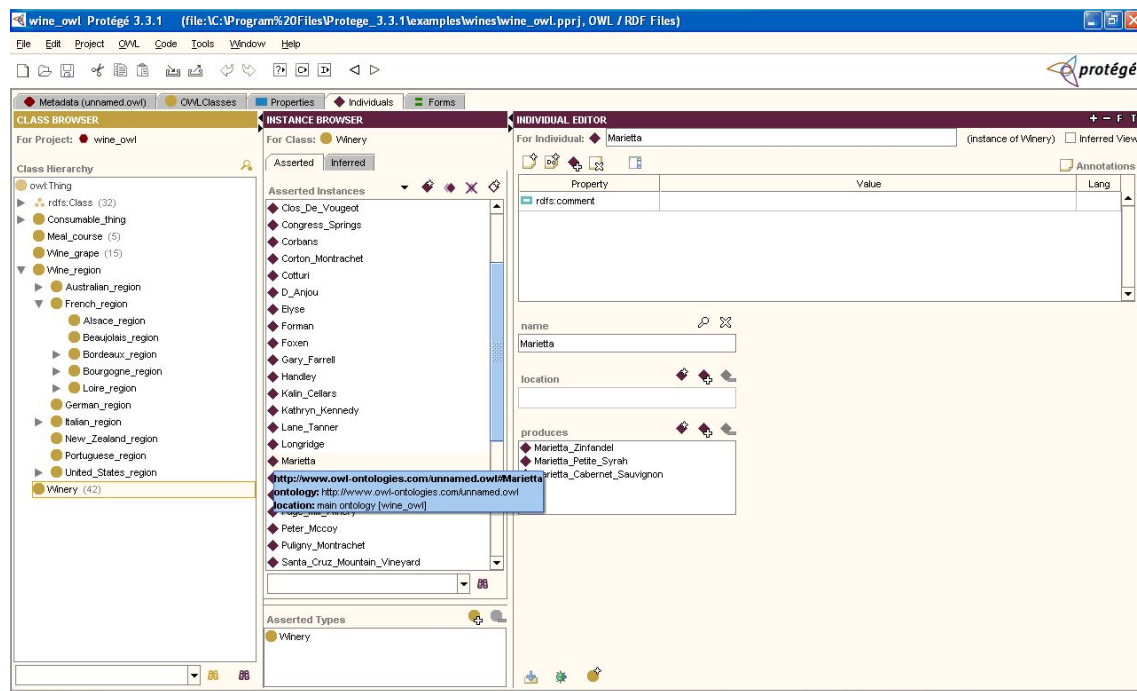
Obrázek 8.3: Karta OWLClasses

Záložka Properties slouží k tvorbě a editování vlastností. Karta je opět rozdělena na dvě části. V levé naleznete hierarchicky uspořádaný seznam vlastností, jež je ještě rozdělen na vlastnosti objektové, datotypové a anotační. Pravá část slouží k úpravě parametrů vybrané vlastnosti. Opět můžeme vkládat poznámky a komentáře, ale především nastavovat omezení vlastností pomocí *rdfs:domain* a *rdfs:range*. Mimo jiné zde můžeme nastavit, zda se jedná o vlastnost symetrickou, transitivní, funkcionální nebo inverzněfunkcionální.



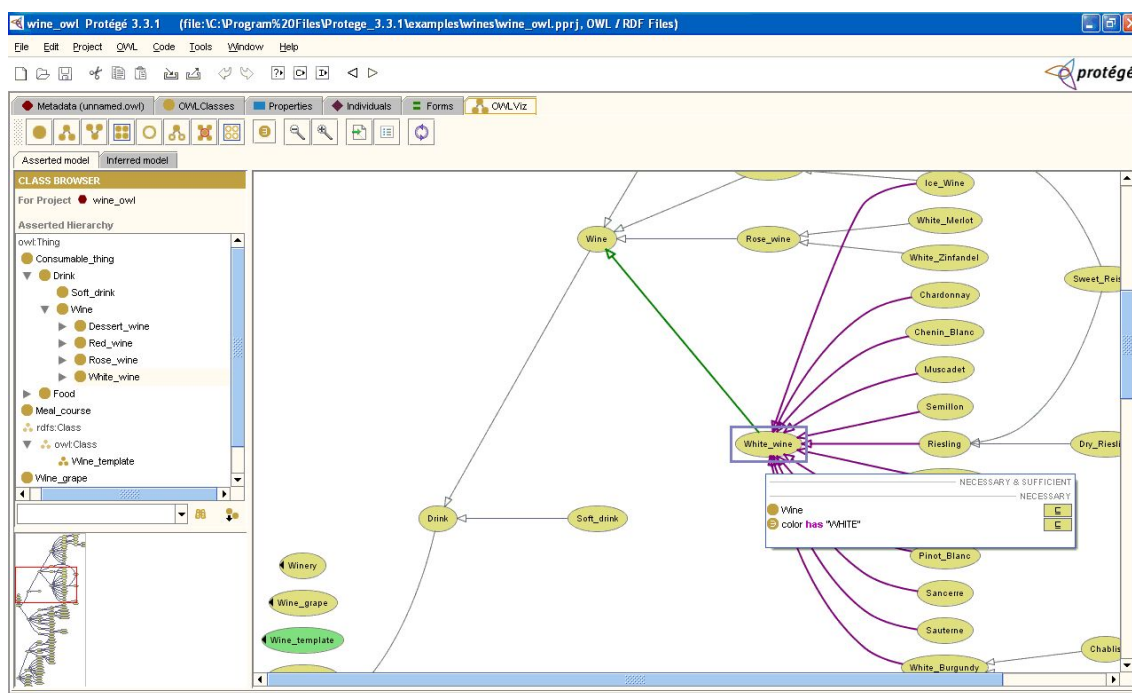
Obrázek 8.4: Karta Properties

Záložka Individuals obsahuje nástroje pro tvorbu a editaci instancí jednotlivých tříd. Karta je rozdělena na tři části. Levá část je obdobně jako u karty Classes umístěn uspořádaný strom tříd. Prostřední část obsahuje seznam instancí právě vybrané třídy. V pravé části se nachází vlastní editor instancí, v němž lze instancím přiřazovat vlastnosti a popisky.



Obrázek 8.5: Karta Individuals

Ke výše zmíněným základním záložkám lze podle potřeb přidávat záložky další. Například záložku OWLViz, která obsahuje funkce obsluhující graf tříd. Karta je rozdělena na dvě části. V první, levé části se opět nachází uspořádaný strom tříd a celkový náhled na graf s vyznačením aktuální pozice v grafu. V pravé části je detail grafu. Další součástí této karty je panel nástrojů. Ten umožňuje přiblížit či oddálit části grafu, export grafu do jiných formátů, ovládat zobrazení grafu apod.



Obrázek 8.6: Karta OWLViz

Kapitola 9

Návrh ontologií

Návrh ontologií je velmi složitý proces. Kvůli jedinečné podobě každé nově navrhované ontologie není jeden „správný“ návod, jak ontologie vytvářet. Dá se ovšem říci, že existuje několik obecně platných pravidel.

Tvorbu ontologického modelu lze rozdělit do těchto kroků, [7]:

- Stanovení rozsahu a cíle ontologie.
- Identifikace entit specifických v dané doméně.
- Uspořádání entit do hierarchie.
- Definice entit.
- Vlastnosti entit.
- Identifikace vztahů.
- Výběr vhodného softwaru.

9.1 Stanovení rozsahu a cíle ontologie

Nezbytným krokem při návrhu ontologie je zodpovězení těchto otázek:

- Z jakých důvodů ontologii vytváříme. – výzkum, trénovací data, uplatnění v praxi...
- Jakému účelu bude ontologie sloužit. – základ pro databázi, doplňková data...
- Komu bude ontologie sloužit (cílová skupina uživatelů).
- Bude-li se ontologie vyvíjet dále v čase. – doplňování nových informací, upřesňování vztahů mezi jednotlivými prvky...
- Z jakých podkladů bude ontologie vycházet. – směrnice, normy...

Výše uvedné otázky, respektive odpovědi na ně, je důležité si rozmyslet, neboť se od nich bude odvíjet celý proces návrhu ontologie.

9.2 Identifikace entit specifických v dané doméně

Před vytvořením vlastní ontologie je nutné identifikovat entity specifické pro danou doménu. Výsledná ontologie by měla obsahovat všechny informace, které jsou nutné, a naopak by neměla obsahovat informace zbytečné nebo s doménou nesouvisející. V případě, že se jedná o entity vyjadřující nějaký stav, je nejprve nutné si je definovat, tj. určit referenční podmínky.

9.3 Uspořádání entit do hierarchie

Další věcí, kterou je třeba určit při definování entit, je roztrídění těchto do tříd, určení jejich vzájemných vztahů a hierarchie. Existuje několik přístupů roztrídění entit do tříd, [7]:

- *Shora-dolů*. Proces začíná definicí nejobecnějších pojmů a poté následuje specializace.
- *Zdola-nahoru*. Proces začíná definicí konkrétních tříd (listy hierarchie) a poté následným sdružováním do obecnějších pojmů.
- *Kombinace*. Kombinace obou přístupů (shora-dolů a zdola-nahoru). Nejdříve se definují nejzásadnější pojmy a poté se generalizují a vhodně specializují.

9.4 Definice entit

Definice entit modelu vychází z kroku 2 (Identifikace entit specifických v dané doméně – 9.2). Je dobré si pro následnou snazší orientaci v modelu ujednotit způsob označování entit a vyvarovat se používání nežádoucích znaků (např. diakritika...).

9.5 Vlastnosti entit

Z předchozího textu (kapitola 2.4) víme, že každá třída obsahuje vlastnosti, které se k ní připojují jako sloty. Obecně existuje několik typů vlastností, [7]:

- *Vnitřní vlastnosti*. Inherentní vlastnosti jsou pro třídu zásadní. Absence těchto vlastností znamená ztrátu identity. Např. umístění a poloha objektu.
- *Vnější vlastnosti*. Např. název objektu.
- *Části*. Mohou být fyzické nebo abstraktní.
- *Vztahy*. Vztahy mezi jednotlivými členy třídy a dalšími položkami.

I zde je dobré ujednotit způsob jejich označení.

9.6 Identifikace vztahů

Hodnota jednoho slotu může rovněž záviset na hodnotě slotu jiného. Tyto vztahy nazýváme „maker“ a „producer“. Jedná se o vztahy vzájemně inverzní. Ukládání vztahů „v obou směrech“ je nadbytečné a zbytečně se tím zvětšuje velikost ontologie. Aplikace dokáží sami s využitím báze znalostí odvodit hodnoty pro tyto druhy vztahů. Ovšem při tvorbě modelu nijak neuškodí mít obě informace k dispozici.

9.7 Výběr vhodného softwaru

Posledním krokem před samotnou tvorbou ontologie je volba vhodného softwaru. Ta závisí na používaném operačním systému, na jazyku, v němž chceme ontologii vytvářet, a konečně i na možnostech reprezentace ontologie.

Podrobnější informace o dostupném softwaru jsou uvedeny v kapitole 7.

Kapitola 10

Ontologie a jejich využití

Ontologie je jedna z možností zápisu struktury dat týkajících se jedné domény. Jedná se tedy o jakýsi obecný formát, který, je-li správně využit, lze použít pro zápis znalostí z jakékoli oblasti lidského vědění. Ovšem je třeba hned říci, že ne všechny znalosti je možné tímto způsobem zapsat.

Najít reálné aplikace je prozatím nesnadný úkol. V současné době je hotových, volně dostupných ontologií jen malé množství. Tuto skutečnost je třeba přičíst faktu, že se jedná o poměrně „mladou“ disciplinu informatiky. Poměrně velké knihovny ontologií můžeme nalézt na internetových stránkách editorů ontologií:

- Editor Protégé.
http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#Other_ontology_formats
- Jazyk DAML.
<http://www.daml.org/ontologies/>
- MapOnto Project.
<http://www.cs.toronto.edu/semanticweb/maponto/ontologies.html>

10.1 Ontologie v geoinformatice, kartografii a geodezii

Pokud se zaměříme na ontologie z oblasti geoinformatiky, kartografie a geodézie, vytvořených ontologií nalezneme jen velmi málo. Jsou to například tyto:

- *Geographic Information Metadata*. Jedná se o ontologii týkající se struktury metadat podle normy ISO 19115. Ontologie je k dispozici na stránkách Protege Wiki v sekci Protege Ontology Library. Bohužel tato není v současnosti dostupná.
- *Geographic Location*. Ontologie popisuje definování polohy pomocí zeměpisné délky (Longitude), zeměpisné šířky (Latitude) a nadmořské výšky (Altitude).
<http://www.daml.org/ontologies/156>

- *Geographic Feature Names*. Ontologie pro geografická jména založená na GEOnet Names Serveru¹ (GNS) agentury NIMA (The National Imagery and Mapping Agency).
<http://www.daml.org/ontologies/346>
- *Longitude, Latitude*. Ontologie popisující souřadnicový systém používaný technologií GPS.
<http://www.daml.org/ontologies/269>
- *Map*. Tato ontologie popisuje mapové vrstvy bez ohledu na zobrazovací systém.
<http://www.daml.org/ontologies/177>

10.2 Návrh ontologií pro geoinformatiku, kartografii a geodezii

Jak již bylo řečeno, možnosti využití ontologií jsou široké. Proto ani nepřekvapí, že lze této techniky uplatnit i v oblasti geoinformatiky, kartografie a geodézie. Jako vhodní kandidáté pro definování nových ontologií se jeví tyto okruhy:

- *Struktura resortu zeměměřictví a katastru*.
Ontologie znázorňující strukturu celého resortu. Výhodou je přehlednost, snadná aktualizace a možnost zobrazení vývoje struktury resortu. Uplatnění spatřuji především na internetových stránkách nebo jako studijní materiál.
- *Geodetické základy*.
Tato ontologie zobrazuje nejen strukturu geodetických základů, ale obsahuje i informace o vlastnostech bodů, číslování, způsobu stabilizace a údržbě. Uplatnění tato ontologie nalezne opět na internetových stránkách nebo jako studijní materiál.
- *Mapové podklady v ČR*.
Ontologie je souhrnem všech informací o existujících mapových podkladech na území České republiky. Jestliže bude do ontologie zapracována i informace o jednotlivých oblastech, kde se které mapové podklady nachází, bude snadné si před vlastní prací v terénu vyhledat, jaké podklady jsou v oblasti k dispozici, a podle toho následně zvolit postup prací. Dále lze ontologii uplatnit na webových stránkách či webových aplikacích.
- *Katalogizace měřících přístrojů*.
Ontologie obsahující informace o geodetických i fotogrammetrických měřících přístrojích. Jedná se v podstatě o katalog historických i současných přístrojů obsahující i technické údaje. Vhodná se jeví zejména pro muzejní nebo studijní účely.
- *Geografická metadata*.
Ontologie popisující strukturu metadat podle normy ISO 19115. Výhodou této ontologie a jejího využívání je v budoucnu stejná forma všech metadat. Metadata budou také obsahovat pouze relevantní informace. Rovněž bude usnadněno programování aplikací, jež budou s metadaty pracovat.

¹<http://geography.usgs.gov>

- *Kartografická zobrazení.*

Ontologie popisující systém existujících kartografických zobrazení včetně jejich vlastností. Ontologie poskytne přehlednou strukturu, v níž bude možné pomocí vhodných nástrojů vyhledávat kartografická zobrazení podle požadovaných vlastností. Další výhodou tohoto způsobu zápisu zobrazení je poměrně snadné doplnění nově definovaných zobrazení. Tato ontologie nalezene opět využití na webových stránkách nebo jako studijní materiál.

- *Souřadnicové systémy.*

Ontologie definující souřadnicové systémy. Tato nám poskytne přehled o možnostech vyjádření polohy místa. Pokud budou zahrnuty i údaje o existujících souřadnicových systémech a jejich využití, bude rovněž možné pomocí této ontologie vyhledávat nejvhodnější zobrazení pro námi zpracovávanou oblast. Stejně jako předchozí podobné ontologie bude také vhodným studijním materiálem a nalezne rovněž uplatnění na Internetu.

- *Značkový klíč.*

Jedná se o ontologii poměrně složitou. Výsledkem by měla být ontologie znázorňující strukturu mapových značek a jejich vlastností. Grafické znázornění této by mělo odrážet různé způsoby tématického rozdělení značek. Z pohledu tématického, grafického nebo podle měřítka mapy. Značkové klíče vytvářené na základě této ontologie by měly být snadno vzájemně porovnatelné a zaměnitelné. Také změna grafického vyjádření jednotlivých značek nebo doplnění nových značek by nemělo být nijak složité. Takové mapové značky jistě naleznou své uplatnění i v různých webových mapových aplikacích.

Kapitola 11

Závěr

Ontologické inženýrství je jednou z možností dalšího rozvoje zpracování informací. V České republice je výzkum této problematiky teprve na počátku.

Z informací získaných touto prací je patrné, že výzkum v oblasti ontologického inženýrství má i nadále smysl rozvíjet. Ontologie mohou v budoucnosti sloužit nejen jako významná součást sémantického webu, ale naleznou uplatnění i v dalších oblastech informatiky. Hlavní možnosti spatřuji především ve znalostních systémech, kde mohou být vhodnou metodou pro utřídění a zpřehlednění znalostí. Ontologie mohou mít své uplatnění rovněž jako součásti nástrojů pro rozhodování a řešení problémů. Od těchto výše uvedených možností aplikace ontologií v praxi se odvíjí i využití ontologií v geoinformatice, popřípadě v kartografii nebo v geodézii.

Je zřejmé, že další vývoj ontologického inženýrství v České republice by měl být veden ve spolupráci s výzkumem v zahraničí. Je třeba zdůraznit, že by neměl být orientován pouze na teoretickou stránku problému, ale i na vlastní tvorbu ontologií.

Literatura

- [1] BERNERS-LEE, T., HENDLER, J., LASSILLA, O. *The Semantic Web*. Scientific American, 2001, vol. 284, no. May, str. 35-43.
<http://www.sciam.com/2001/0501issue/0501berners/lee.html>
- [2] SKLENÁK, V. *Metadata, sémantika a sémantický web*. INFORUM 2004. Praha, 2004.
http://www.inforum.cz/pdf/2004/Sklenak_Vilem1.pdf
- [3] SKLENÁK, V. *Sémantický web*. Vysoká škola ekonomická, fakulta informatiky a statistiky. Praha.
- [4] SVÁTEK, V. *Ontologie a WWW*. Proc. Datakon 2002. Brno, 2002.
<http://nb.vse.cz/svatek/onto-www.pdf>
- [5] STUHLÍK, R. *Diplomová práce - Ontologie a sémantický web*. Vysoké učení technické v Brně. Brno, 2007.
<http://www.fit.vutbr.cz/study/DP/rpfile.php?id=5004>
- [6] BRICKLEY, D., GUHA, R. V. *RDF Vocabulary Description Language 1.0* RDF Schema W3C. 2004.
<http://www.w3.org/TR/rdf-scheme>
- [7] ADÁMEK, L. *Diplomová práce - Analýza sémantických technologií pro implementaci systémů v oblasti hodnocení ekologického stavu povrchových vod*. Fakulta informatiky, Masarykovy univerzity v Brně. Brno, 2008.
<http://lukasadamek.ic.cz/index.html>
- [8] <http://protege.stanford.edu>

Seznam zkratek

Zkratka	Význam
API	Application Programming Interface
BNF	Backus–Naur Form
COE	CMap Tools Ontology Editor
DAML	DARPA Agent Mark-up Language
DARPA	The Defense Advanced Research Projects Agency
DIG	The DL Implementation Group
DL	Description Logics
DOVE	DERI Ontology Management Environment
DTD	Document Type Definition
GNS	GEOnet Names Server
GPS	Global Positioning System
HTML	HyperText Markup Language
IHMC	Institute for Human & Machine Cognition
JOE	Java Ontology Editor
KAON	Karlsruhe Ontology
KIF	Knowledge Interchange Format
KSL	Knowledge System Laboratory
LISP	List Processing
N3	Notation 3
NID	Namespace Identifier
NIMA	The National Imagery and Mapping Agency
NSS	Namespace Specific String
OCML	Operational Conceptual Modelling Language
OIL	Ontology Inference Layer
OKBC	Open Knowledge Base Connectivity
OMWG	Ontology Management Working Group
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RFC	Request for Comments
RMI	Remote Method Invocation

SHOE	Simple HTML Ontology Extension
SKOS	Simple Knowledge Organisation Systems
SPARQL	SPARQL Protocol and RDF Query Language
SWRL	Semantic Web Rule Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
XOL	Extensible Ontology Language

Seznam tabulek

3.1	Konstruktory logických výrazů	18
3.2	Axiomy jazyka DAML+OIL	18
3.3	Syntaxe konstruktorů	20
3.4	Sémantika konstruktorů	20
4.1	Typové literály	26
5.1	RDFS třídy	28
6.1	Omezení hodnotou	35
6.2	Omezení kardinality	35
7.1	Přehled editorů ontologií	41

Seznam obrázků

1.1	Typy metadat	6
1.2	Vennův diagram vztahu URI, URN, URL	8
4.1	Trojice subjekt – predikát – objekt	24
8.1	Logo Protégé	46
8.2	Karta Metadata	48
8.3	Karta OWLClasses	49
8.4	Karta Properties	50
8.5	Karta Individuals	51
8.6	Karta OWLViz	52

Příloha A

Graf ukázkové ontologie wine.owl

Příloha B

Zdrojový kód ukázkové ontologie wine.owl

```
< ?xml version="1.0"? >
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
xml:base="http://www.owl-ontologies.com/unnamed.owl" >
<owl:Ontology rdf:about="" />
<owl:Class rdf:ID="Light_meat_fowl" >
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Light meat fowl</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Fowl" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="St_Emillion_region" >
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Bordeaux_region" />
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >St Emillion region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Meursault_region" >
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Meursault region</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Bourgogne_region" />
  </rdfs:subClassOf>
</owl:Class>
```

```

    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Sweet_dessert">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Dessert"/>
    </rdfs:subClassOf>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sweet dessert</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Wine_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Wine region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Tomato-based_food">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Tomato-based food</rdfs:label>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Food"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="New_Zealand_region">
    <rdfs:subClassOf rdf:resource="#Wine_region"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >New Zealand region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Chianti_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chianti region</rdfs:label>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Italian_region"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Central_coast_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Central coast region</rdfs:label>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="United_States_region"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Seafood">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Food"/>

```

```

    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="French_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >French region</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Wine_region" />
</owl:Class>
<owl:Class rdf:ID="White_wine">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty>
                <owl:FunctionalProperty rdf:ID="color" />
            </owl:onProperty>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Wine" />
    </rdfs:subClassOf>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >White wine</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Portuguese_region">
    <rdfs:subClassOf rdf:resource="#Wine_region" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Portuguese region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="White_Bordeaux">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty>
                <owl:FunctionalProperty rdf:about="#color" />
            </owl:onProperty>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#White_wine" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >White Bordeaux</rdfs:label>
</owl:Class>

```

```

<owl:Class rdf:ID="Sauterne_region">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Bordeaux_region"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Sauterne region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Fish">
  <rdfs:subClassOf rdf:resource="#Seafood"/>
</owl:Class>
<owl:Class rdf:ID="Non-sweet_fruit">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Non-sweet fruit</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Fruit"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Non-bland_fish">
  <rdfs:subClassOf rdf:resource="#Fish"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Non-bland fish</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="White_Merlot">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    White Merlot</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#color"/>
      </owl:onProperty>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        ROSdž</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Rose_wine"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Beaujolais_region">
  <rdfs:subClassOf rdf:resource="#French_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Beaujolais region</rdfs:label>

```

```

</owl:Class>
<owl:Class rdf:ID="Red_wine">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#color"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Wine"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Red wine</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Edna_Valley_region">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#United_States_region"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Edna Valley region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Drink">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Consumable_thing"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Dessert_wine">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Dessert wine</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="sugar"/>
      </owl:onProperty>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >SWEET</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>

```

```

        <owl:Class rdf:about="#Wine" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Arroyo_Grande_region">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#United_States_region" />
    </rdfs:subClassOf>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Arroyo Grande region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Pasta_with_spicy_red_sauce">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pasta with spicy red sauce</rdfs:label>
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Pasta_with_red_sauce" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Bordeaux_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Bordeaux region</rdfs:label>
    <rdfs:subClassOf rdf:resource="#French_region" />
</owl:Class>
<owl:Class rdf:ID="German_region">
    <rdfs:subClassOf rdf:resource="#Wine_region" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >German region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Alsace_region">
    <rdfs:subClassOf rdf:resource="#French_region" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Alsace region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Soft_drink">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Soft drink</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Drink" />
</owl:Class>
<owl:Class rdf:about="#Consumable_thing">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Consumable thing</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Meat">

```



```

    <rdfs:subClassOf>
      <owl:Class rdf:about="#Food" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Paulliac_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Paulliac region</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Bordeaux_region" />
  </owl:Class>
  <owl:Class rdf:ID="Cotes_d_Or_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cotes d'Or region</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Bourgogne_region" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Sancerre_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sancerre region</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Loire_region" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Pasta_with_heavy_cream_sauce">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pasta with heavy cream sauce</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Pasta_with_white_sauce" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Australian_region">
    <rdfs:subClassOf rdf:resource="#Wine_region" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Australian region</rdfs:label>
  </owl:Class>
  <owl:Class rdf:ID="Tours_region">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Tours region</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Loire_region" />
    </rdfs:subClassOf>
  </owl:Class>

```

```

<owl:Class rdf:ID="Santa_Cruz_Mountains_region">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#United_States_region"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Santa Cruz Mountains region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Non-spicy_red_meat">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Non-spicy red meat</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Red_meat"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Wine">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    A wine class represents all possible wines</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Drink"/>
</owl:Class>
<owl:Class rdf:ID="Bland_fish">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Bland fish</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Fish"/>
</owl:Class>
<owl:Class rdf:ID="Pays_Nantes_region">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Pays Nantes region</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Loire_region"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Food">
  <rdfs:subClassOf rdf:resource="#Consumable_thing"/>
</owl:Class>
<owl:Class rdf:ID="Napa_Valley_region">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Napa Valley region</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#United_States_region"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mendocino_region">

```

```

<rdfs:subClassOf>
  <owl:Class rdf:about="#United_States_region"/>
</rdfs:subClassOf>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mendocino region</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Rose_wine">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >ROSDž"</owl:hasValue>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#color"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Rose wine</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Sweet_fruit">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Fruit"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sweet fruit</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Pasta_with_light_cream_sauce">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Pasta_with_white_sauce"/>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pasta with light cream sauce</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="South_Australian_region">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >South Australian region</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Australian_region"/>
</owl:Class>
<owl:Class rdf:ID="Dark_meat_fowl">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Dark meat fowl</rdfs:label>

```

```

    <rdfs:subClassOf>
      <owl:Class rdf:about="#Fowl" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Winery" />
  <owl:Class rdf:about="#Pasta_with_white_sauce">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Pasta" />
    </rdfs:subClassOf>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pasta with white sauce</rdfs:label>
  </owl:Class>
  <owl:Class rdf:about="#Red_meat">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Red meat</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Meat" />
  </owl:Class>
  <owl:Class rdf:ID="Wine_template">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Wine template</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Class" />
  </owl:Class>
  <owl:Class rdf:about="#United_States_region">
    <rdfs:subClassOf rdf:resource="#Wine_region" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >United States region</rdfs:label>
  </owl:Class>
  <owl:Class rdf:ID="White_meat">
    <rdfs:subClassOf rdf:resource="#Meat" />
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >White meat</rdfs:label>
  </owl:Class>
  <owl:Class rdf:ID="Shellfish">
    <rdfs:subClassOf rdf:resource="#Seafood" />
  </owl:Class>
  <owl:Class rdf:ID="Cheese_Nuts_dessert">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cheese/Nuts dessert</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Dessert" />
    </rdfs:subClassOf>
  </owl:Class>

```

```

<owl:Class rdf:about="#Bourgogne_region">
  <rdfs:subClassOf rdf:resource="#French_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Bourgogne region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Margaux_region">
  <rdfs:subClassOf rdf:resource="#Bordeaux_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Margaux region</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Pasta">
  <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>
<owl:Class rdf:about="#Fruit">
  <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>
<owl:Class rdf:ID="Wine_grape">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Wine grape</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Spicy_red_meat">
  <rdfs:subClassOf rdf:resource="#Red_meat"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Spicy red meat</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Cotes_Chalonaise_region">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cotes Chalonaise region</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Bourgogne_region"/>
</owl:Class>
<owl:Class rdf:ID="Pasta_with_non-spicy_red_sauce">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pasta with non-spicy red sauce</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Pasta_with_red_sauce"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Loire_region">
  <rdfs:subClassOf rdf:resource="#French_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Loire region</rdfs:label>
</owl:Class>

```

```

<owl:Class rdf:about="#Italian_region">
  <rdfs:subClassOf rdf:resource="#Wine_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Italian region</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Meal_course">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Meal course</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Dessert">
  <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>
<owl:Class rdf:about="#Fowl">
  <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>
<owl:Class rdf:ID="Santa_Barbara_region">
  <rdfs:subClassOf rdf:resource="#United_States_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Santa Barbara region</rdfs:label>
</owl:Class>
<owl:Class rdf:about="#Pasta_with_red_sauce">
  <rdfs:subClassOf rdf:resource="#Pasta"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pasta with red sauce</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Sonoma_region">
  <rdfs:subClassOf rdf:resource="#United_States_region"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sonoma region</rdfs:label>
</owl:Class>
<owl:ObjectProperty rdf:ID="grape">
  <rdfs:domain rdf:resource="#Wine"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="area">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >A specific area (such as Napa) where a wine comes from</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="region"/>
<owl:ObjectProperty rdf:ID="drink">
  <rdfs:domain rdf:resource="#Meal_course"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="produces">

```

```

<owl:inverseOf>
  <owl:FunctionalProperty rdf:ID="maker" />
</owl:inverseOf>
<rdfs:domain rdf:resource="#Winery" />
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>This slot contains the wines produced by a particular winery.
The maker of a Wine is its inverse slot.</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="best_wineries">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>best wineries</rdfs:label>
  <rdfs:domain rdf:resource="#Wine_template" />
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:about="#sugar">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain rdf:resource="#Wine" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="flavor">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain rdf:resource="#Wine" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="location">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty" />
  <rdfs:domain rdf:resource="#Winery" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="Course">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="body">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain rdf:resource="#Wine" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#color">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain rdf:resource="#Wine" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="name">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">

```

```

        <owl:Class rdf:about="#Winery"/>
        <owl:Class rdf:about="#Consumable_thing"/>
        <owl:Class rdf:about="#Wine_grape"/>
    </owl:unionOf>
</owl:Class>
</rdfs:domain>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="food">
    <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Meal_course"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="tannin_level">
    <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >tannin level</rdfs:label>
    <rdfs:domain rdf:resource="#Red_wine"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#maker">
    <owl:inverseOf rdf:resource="#produces"/>
    <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Wine"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The maker of a wine (a Winery).
    This slot has an iinverse - the slot produces at the Winery class</rdfs:comment>
</owl:FunctionalProperty>
<Pasta_with_spicy_red_sauce rdf:ID="Fra_Diavolo">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Fra Diavolo</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Fra Diavolo</name>
</Pasta_with_spicy_red_sauce>
<Bland_fish rdf:ID="Scrod">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Scrod</name>
</Bland_fish>
<Wine_grape rdf:ID="Pinot_Blanc_grape">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pinot Blanc grape</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pinot Blanc grape</name>
</Wine_grape>

```



```

<Wine_template rdf:ID="Chardonnay">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Wine_template rdf:ID="Red_Burgundy">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Red_Burgundy</rdfs:label>
</Wine_template>
<Light_meat_fowl rdf:ID="Turkey">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Turkey</name>
</Light_meat_fowl>
<Dry_Riesling rdf:ID="Corbans_Dry_White_Riesling">
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >OFF-DRY</sugar>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Corbans Dry White Riesling</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Corbans Dry White Riesling</rdfs:label>
  <maker>
    <Winery rdf:ID="Corbans">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Corbans</name>
      <produces>
        <Sauvignon_Blanc rdf:ID="Corbans_Sauvignon_Blanc">
          <maker rdf:resource="#Corbans"/>

```

```

    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Corbans Sauvignon Blanc</name>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >STRONG</flavor>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Corbans Sauvignon Blanc</rdfs:label>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
  </Sauvignon_Blanc>
</produces>
<produces rdf:resource="#Corbans_Dry_White_Riesling"/>
<produces>
  <Sauvignon_Blanc rdf:ID="Corbans_Private_Bin_Sauvignon_Blanc">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Corbans Private Bin Sauvignon Blanc</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Corbans Private Bin Sauvignon Blanc</name>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >STRONG</flavor>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >FULL</body>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <maker rdf:resource="#Corbans"/>
  </Sauvignon_Blanc>
</produces>
</Winery>
</maker>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MODERATE</flavor>
<color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>WHITE</color>
</Dry_Riesling>
<Wine_template rdf:ID="Petite_Syrah">
  <rdfs:subClassOf>
    <owl:Restriction>

```

```

        <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
        <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#Red_wine"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Petite Syrah</rdfs:label>
</Wine_template>
<Wine_template rdf:ID="Riesling">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#color"/>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Wine_grape rdf:ID="Petite_Syrah_grape">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Petite Syrah grape</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Petite Syrah grape</name>
</Wine_grape>
<Winery rdf:ID="Bancroft">
    <produces>
        <Chardonnay rdf:ID="Bancroft_Chardonnay">
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Bancroft Chardonnay</rdfs:label>
            <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >DRY</sugar>
            <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >MODERATE</flavor>
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Bancroft Chardonnay</name>
            <maker rdf:resource="#Bancroft"/>
            <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >MEDIUM</body>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</color>
        </Chardonnay>
    </produces>

```

```

</produces>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Bancroft</name>
</Winery>
<Wine_template rdf:ID="Sauterne">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Chenin_Blanc rdf:ID="Ventana_Chenin_Blanc">
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >OFF-DRY</sugar>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Ventana Chenin Blanc</name>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</flavor>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Ventana Chenin Blanc</rdfs:label>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <maker>
    <Winery rdf:ID="Ventana">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Ventana</name>
      <produces rdf:resource="#Ventana_Chenin_Blanc"/>
    </Winery>
  </maker>
</Chenin_Blanc>

```

```

<Sweet_dessert rdf:ID="Pie">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pie</name>
</Sweet_dessert>
<Winery rdf:ID="Saucelito_Canyon">
  <produces>
    <Red_Zinfandel rdf:ID="Saucelito_Canyon_Zinfandel">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Saucelito Canyon Zinfandel</rdfs:label>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Saucelito Canyon Zinfandel</name>
      <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >MEDIUM</body>
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <maker rdf:resource="#Saucelito_Canyon"/>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >MODERATE</flavor>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
    </Red_Zinfandel>
  </produces>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Saucelito Canyon</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Saucelito Canyon</rdfs:label>
</Winery>
<Shellfish rdf:ID="Oysters">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Oysters</name>
</Shellfish>
<Chardonnay rdf:ID="Mount_Eden_Vineyard_Edna_Valley_Chardonnay">
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</flavor>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mount Eden Vineyard Edna Valley Chardonnay</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mount Eden Vineyard Edna Valley Chardonnay</rdfs:label>
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>

```

```

<maker>
  <Winery rdf:ID="Mount_Eden_Vineyard">
    <produces>
      <Pinot_Noir rdf:ID="Mount_Eden_Vineyard_Estate_Pinot_Noir">
        <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >DRY</sugar>
        <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</color>
        <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >FULL</body>
        <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >STRONG</flavor>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Mount Eden Vineyard Estate Pinot Noir</name>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Mount Eden Vineyard Estate Pinot Noir</rdfs:label>
        <maker rdf:resource="#Mount_Eden_Vineyard"/>
      </Pinot_Noir>
    </produces>
    <produces rdf:resource="#Mount_Eden_Vineyard_Edna_Valley_Chardonnay"/>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mount Eden Vineyard</name>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mount Eden Vineyard</rdfs:label>
  </Winery>
</maker>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
</Chardonnay>
<Red_Zinfandel rdf:ID="Cotturi_Zinfandel">
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cotturi Zinfandel</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cotturi Zinfandel</rdfs:label>
<maker>
  <Winery rdf:ID="Cotturi">
    <produces rdf:resource="#Cotturi_Zinfandel"/>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cotturi</name>
  </Winery>

```

```

</maker>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>FULL</body>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>STRONG</flavor>
</Red_Zinfandel>
<Winery rdf:ID="D_Anjou">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >D_Anjou</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >D_Anjou</name>
</Winery>
<Non-bland_fish rdf:ID="Swordfish">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Swordfish</name>
</Non-bland_fish>
<Winery rdf:ID="Sterling_Vineyards">
  <location rdf:resource="#Napa_Valley_region"/>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Sterling Vineyards</name>
  <produces>
    <Red_Merlot rdf:ID="wines-current_00049">
      <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >FULL</body>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >STRONG</flavor>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Sterling Merlot</name>
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
      <maker rdf:resource="#Sterling_Vineyards"/>
    </Red_Merlot>
    <grape>
      <Wine_grape rdf:ID="Merlot_grape">
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Merlot grape</name>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Merlot grape</rdfs:label>
      </Wine_grape>
    </grape>
  </produces>
</Winery>

```

```

    </grape>
  </Red_Merlot>
</produces>
<produces>
  <Cabernet_Sauvignon rdf:ID="wines-current_00051">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sterling Cabernet Sauvignon</name>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <maker rdf:resource="#Sterling_Vineyards" />
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
    <grape>
      <Wine_grape rdf:ID="Cabernet_Sauvignon_grape">
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Cabernet Sauvignon grape</name>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Cabernet Sauvignon grape</rdfs:label>
      </Wine_grape>
    </grape>
  </Cabernet_Sauvignon>
</produces>
<produces>
  <Chardonnay rdf:ID="wines-current_00050">
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <grape>
      <Wine_grape rdf:ID="Chardonnay_grape">
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chardonnay grape</rdfs:label>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chardonnay grape</name>
      </Wine_grape>
    </grape>
    <maker rdf:resource="#Sterling_Vineyards" />
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sterling Chardonnay</name>
  </Chardonnay>
</produces>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Sterling Vineyards</rdfs:label>
</Winery>

```



```

<Sweet_fruit rdf:ID="Banana">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Banana</name>
</Sweet_fruit>
<Wine_grape rdf:ID="Chenin_Blanc_grape">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chenin Blanc grape</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chenin Blanc grape</rdfs:label>
</Wine_grape>
<Port rdf:ID="Taylor_Port">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Taylor Port</rdfs:label>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >SWEET</sugar>
  <maker>
    <Winery rdf:ID="Taylor">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Taylor</name>
      <produces rdf:resource="#Taylor_Port"/>
    </Winery>
  </maker>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Taylor Port</name>
</Port>
<Wine_template rdf:ID="Beaujolais">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
</Wine_template>
<Winery rdf:ID="Puligny_Montrachet">
  <produces>
    <White_Burgundy rdf:ID="Puligny_Montrachet_White_Burgundy">
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</color>

```

```

    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Puligny Montrachet White Burgundy</name>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Puligny Montrachet White Burgundy</rdfs:label>
    <maker rdf:resource="#Puligny_Montrachet"/>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MODERATE</flavor>
  </White_Burgundy>
</produces>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Puligny Montrachet</rdfs:label>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Puligny Montrachet</name>
</Winery>
<Chenin_Blanc rdf:ID="Foxen_Chenin_Blanc">
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >FULL</body>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</flavor>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <maker>
    <Winery rdf:ID="Foxen">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Foxen</name>
      <produces rdf:resource="#Foxen_Chenin_Blanc"/>
    </Winery>
  </maker>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Foxen Chenin Blanc</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Foxen Chenin Blanc</name>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DRY</sugar>
</Chenin_Blanc>
<Sweet_fruit rdf:ID="Grape">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Grape</name>

```

```

</Sweet_fruit>
<Wine_template rdf:ID="Cabernet_Franc">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cabernet Franc</rdfs:label>
</Wine_template>
<Non-spicy_red_meat rdf:ID="Steak">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Steak</name>
</Non-spicy_red_meat>
<Red_Zinfandel rdf:ID="Marietta_Zinfandel">
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
  <maker>
    <Winery rdf:ID="Marietta">
      <produces rdf:resource="#Marietta_Zinfandel"/>
      <produces>
        <Petite_Syrah rdf:ID="Marietta_Petite_Syrah">
          <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >MODERATE</flavor>
          <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Marietta Petite Syrah</name>
          <maker rdf:resource="#Marietta"/>
          <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</color>
          <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >DRY</sugar>
          <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >MEDIUM</body>
          <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Marietta Petite Syrah</rdfs:label>
        </Petite_Syrah>
      </produces>
    </produces>
    <Cabernet_Sauvignon rdf:ID="Marietta_Cabernet_Sauvignon">

```

```

    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Marietta Cabernet Sauvignon</rdfs:label>
    <maker rdf:resource="#Marietta"/>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MODERATE</flavor>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Marietta Cabernet Sauvignon</name>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
  </Cabernet_Sauvignon>
</produces>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Marietta</name>
</Winery>
</maker>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Marietta Zinfandel</rdfs:label>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MEDIUM</body>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Marietta Zinfandel</name>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MODERATE</flavor>
</Red_Zinfandel>
<Bland_fish rdf:ID="Halibut">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Halibut</name>
</Bland_fish>
<Meal_course rdf:ID="wines-current_00032">
  <drink>
    <Wine_template rdf:ID="Port">
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >RED</owl:hasValue>
          <owl:onProperty rdf:resource="#color"/>

```

```

        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >SWEET</owl:hasValue>
            <owl:onProperty rdf:resource="#sugar" />
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Dessert_wine" />
    <rdfs:subClassOf rdf:resource="#Red_wine" />
</Wine_template>
</drink>
<food>
    <Cheese_Nuts_dessert rdf:ID="Nuts">
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Nuts</name>
    </Cheese_Nuts_dessert>
</food>
</Meal_course>
<Wine_template rdf:ID="Sweet_Reisling">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >WHITE</owl:hasValue>
            <owl:onProperty rdf:resource="#color" />
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#sugar" />
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >SWEET</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#grape" />
            <owl:hasValue>
                <Wine_grape rdf:ID="Riesling_grape">
                    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                        >Riesling grape</rdfs:label>

```

```

        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Riesling grape</name>
    </Wine_grape>
</owl:hasValue>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#body" />
        <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >FULL</owl:hasValue>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#Dessert_wine" />
<rdfs:subClassOf rdf:resource="#Riesling" />
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Sweet Reisling</rdfs:label>
</Wine_template>
<Cabernet_Sauvignon rdf:ID="Forman_Cabernet_Sauvignon">
    <maker>
        <Winery rdf:ID="Forman">
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Forman</name>
            <produces>
                <Chardonnay rdf:ID="Forman_Chardonnay">
                    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >FULL</body>
                    <maker rdf:resource="#Forman" />
                    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >Forman Chardonnay</rdfs:label>
                    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >MODERATE</flavor>
                    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >Forman Chardonnay</name>
                    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >DRY</sugar>
                    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >WHITE</color>
                </Chardonnay>
            </produces>
            <produces rdf:resource="#Forman_Cabernet_Sauvignon" />
        </Winery>
    </maker>

```

```

</maker>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Forman Cabernet Sauvignon</name>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>STRONG</flavor>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MEDIUM</body>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Forman Cabernet Sauvignon</rdfs:label>
<color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>RED</color>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
</Cabernet_Sauvignon>
<Red_Merlot rdf:ID="Gary_Farrell_Merlot">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Gary Farrell Merlot</name>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DRY</sugar>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Gary Farrell Merlot</rdfs:label>
  <maker>
    <Winery rdf:ID="Gary_Farrell">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Gary Farrell</rdfs:label>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Gary Farrell</name>
      <produces rdf:resource="#Gary_Farrell_Merlot"/>
    </Winery>
  </maker>
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</flavor>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
</Red_Merlot>
<Winery rdf:ID="Chateau_Margaux_Winery">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Margaux Winery</rdfs:label>
  <produces>
    <Margaux rdf:ID="Chateau_Margaux">

```

```

    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Margaux</name>
    <maker rdf:resource="#Chateau_Margaux_Winery"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Margaux</rdfs:label>
  </Margaux>
</produces>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Chateau Margaux</name>
</Winery>
<Spicy_red_meat rdf:ID="Garlicky_roast">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Garlicky roast</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Garlicky roast</rdfs:label>
</Spicy_red_meat>
<Spicy_red_meat rdf:ID="Beef_curry">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Beef curry</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Beef curry</rdfs:label>
</Spicy_red_meat>
<Sweet_Reisling rdf:ID="Schloss_Rothermel_Trochenbierenauslese_Riesling">
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >FULL</body>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >STRONG</flavor>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >SWEET</sugar>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Schloss Rothermel Trochenbierenauslese Riesling</name>
  <maker>
    <Winery rdf:ID="Schloss_Rothermel">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Schloss Rothermel</name>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Schloss Rothermel</rdfs:label>
      <location rdf:resource="#German_region"/>
    </Winery>
  </maker>

```



```

        <produces rdf:resource="#Schloss.Rothermel.Trochenbierenauslese.Riesling"/>
    </Winery>
</maker>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Schloss Rothermel Trochenbierenauslese Riesling</rdfs:label>
<grape rdf:resource="#Riesling_grape"/>
</Sweet_Reisling>
<Winery rdf:ID="Santa_Cruz_Mountain_Vineyard">
    <location rdf:resource="#Santa_Cruz_Mountains_region"/>
    <produces>
        <Cabernet_Sauvignon rdf:ID="Santa_Cruz_Mountain_Vineyard_Cabernet_Sauvignon">
            <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >STRONG</flavor>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</color>
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Santa Cruz Mountain Vineyard Cabernet Sauvignon</name>
            <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >DRY</sugar>
            <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >FULL</body>
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Santa Cruz Mountain Vineyard Cabernet Sauvignon</rdfs:label>
            <maker rdf:resource="#Santa_Cruz_Mountain_Vineyard"/>
        </Cabernet_Sauvignon>
    </produces>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Santa Cruz Mountain Vineyard</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Santa Cruz Mountain Vineyard</name>
</Winery>
<Wine_template rdf:ID="Dry_Riesling">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Dry Riesling</rdfs:label>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</owl:hasValue>
            <owl:onProperty rdf:resource="#color"/>
        </owl:Restriction>
    </rdfs:subClassOf>

```

```

    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Riesling" />
</Wine_template>
<Winery rdf:ID="Elyse">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Elyse</name>
  <produces>
    <Red_Zinfandel rdf:ID="Elyse_Zinfandel">
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >MODERATE</flavor>
      <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >FULL</body>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Elyse Zinfandel</name>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Elyse Zinfandel</rdfs:label>
      <maker rdf:resource="#Elyse" />
    </Red_Zinfandel>
  </produces>
</Winery>
<Sweet_fruit rdf:ID="Mixed_fruit">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mixed fruit</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mixed fruit</rdfs:label>
</Sweet_fruit>
<Winery rdf:ID="Handley">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Handley</name>
</Winery>
<St_Emillion rdf:ID="Chateau_Cheval_Blanc_St_Emilion">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Cheval Blanc St Emilion</name>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Cheval Blanc St Emilion</rdfs:label>
</St_Emillion>

```

```

<Cheese_Nuts_dessert rdf:ID="Cheese">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cheese</name>
</Cheese_Nuts_dessert>
<Meal_course rdf:ID="wines-current_00035">
  <food>
    <Light_meat_fowl rdf:ID="Grilled_chicken">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Grilled chicken</name>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Grilled chicken</rdfs:label>
    </Light_meat_fowl>
  </food>
  <drink rdf:resource="#Beaujolais"/>
</Meal_course>
<Chardonnay rdf:ID="Mountadam_Chardonnay">
  <maker>
    <Winery rdf:ID="Mountadam">
      <produces>
        <Pinot_Noir rdf:ID="Mountadam_Pinot_Noir">
          <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >MEDIUM</body>
          <maker rdf:resource="#Mountadam"/>
          <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >RED</color>
          <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >DRY</sugar>
          <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >MODERATE</flavor>
          <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Mountadam Pinot Noir</rdfs:label>
          <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Mountadam Pinot Noir</name>
        </Pinot_Noir>
      </produces>
      <produces rdf:resource="#Mountadam_Chardonnay"/>
    </produces>
    <Dry_Riesling rdf:ID="Mountadam_Riesling">
      <maker rdf:resource="#Mountadam"/>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Mountadam Riesling</name>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

    >Mountadam Riesling</rdfs:label>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DELICATE</flavor>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
  </Dry_Riesling>
</produces>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mountadam</name>
</Winery>
</maker>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mountadam Chardonnay</rdfs:label>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>STRONG</flavor>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>FULL</body>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mountadam Chardonnay</name>
<color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>WHITE</color>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
</Chardonnay>
<Pauillac rdf:ID="Chateau_Lafite_Rothschild_Pauillac">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Lafite Rothschild Pauillac</name>
  <grape rdf:resource="#Cabernet_Sauvignon_grape"/>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DRY</sugar>
  <tannin_level rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</tannin_level>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >STRONG</flavor>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Lafite Rothschild Pauillac</rdfs:label>

```

```

<maker>
  <Winery rdf:ID="Chateau_Lafite_Rothschild">
    <produces rdf:resource="#Chateau_Lafite_Rothschild_Pauillac"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Lafite Rothschild</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Lafite Rothschild</name>
  </Winery>
</maker>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>FULL</body>
</Pauillac>
<Winery rdf:ID="Kathryn_Kennedy">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Kathryn Kennedy</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Kathryn Kennedy</rdfs:label>
</Winery>
<Wine_template rdf:ID="Pouilly-Fuisse">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf>
  <Wine_template rdf:ID="White_Burgundy">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >White Burgundy</rdfs:label>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >WHITE</owl:hasValue>
        <owl:onProperty rdf:resource="#color"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#White_wine"/>
  </Wine_template>
</rdfs:subClassOf>
</Wine_template>
<Wine_template rdf:ID="Muscadet">

```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#color"/>
    <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#color"/>
    <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Wine_template rdf:ID="Red_Bordeaux">
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
  <best_wineries>
    <Winery rdf:ID="wines_00089">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chateau Mouton-Rothschild</name>
    </Winery>
  </best_wineries>
  <best_wineries rdf:resource="#Chateau_Lafite_Rothschild"/>
  <best_wineries rdf:resource="#Chateau_Margaux_Winery"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The class of all Bordeaux wines</rdfs:comment>
  <best_wineries>
    <Winery rdf:ID="wines_00087">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chateau Latour</name>
    </Winery>
  </best_wineries>
</best_wineries>

```

```

    <Winery rdf:ID="wines_00088">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chateau Haut-Brion</name>
    </Winery>
  </best_wineries>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Red Bordeaux</rdfs:label>
</Wine_template>
<Wine_template rdf:ID="Red_Zinfandel">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Red Zinfandel</rdfs:label>
</Wine_template>
<Shellfish rdf:ID="Clams">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Clams</name>
</Shellfish>
<Wine_template rdf:ID="Pauillac">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <Wine_template rdf:ID="Medoc">
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#color"/>
          <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</owl:hasValue>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf rdf:resource="#Red_Bordeaux"/>
    </Wine_template>
  </rdfs:subClassOf>

```

```

    </Wine_template>
  </rdfs:subClassOf>
</Wine_template>
<Meal_course rdf:ID="wines-current_00029">
  <drink rdf:resource="#Red_Bordeaux"/>
  <food>
    <Red_meat rdf:ID="Lamb">
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Lamb</name>
    </Red_meat>
  </food>
</Meal_course>
<Wine_template rdf:ID="St._Emillion">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >St. Emillion</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_Bordeaux"/>
</Wine_template>
<Non-sweet_fruit rdf:ID="Apple">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Apple</name>
</Non-sweet_fruit>
<Wine_template rdf:ID="Pinot_Noir">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pinot Noir</rdfs:label>
</Wine_template>
<Sauvignon_Blanc rdf:ID="Selaks_Sauvignon_Blanc">
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```



```

>WHITE</color>
<maker>
  <Winery rdf:ID="Selaks">
    <produces>
      <Ice_Wine rdf:ID="Selaks_Ice_Wine">
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Selaks Ice Wine</name>
        <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >MEDIUM</body>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Selaks Ice Wine</rdfs:label>
        <maker rdf:resource="#Selaks" />
        <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >WHITE</color>
        <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >SWEET</sugar>
        <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >MODERATE</flavor>
      </Ice_Wine>
    </produces>
    <produces rdf:resource="#Selaks_Sauvignon_Blanc" />
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Selaks</name>
  </Winery>
</maker>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MEDIUM</body>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MODERATE</flavor>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Selaks Sauvignon Blanc</rdfs:label>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Selaks Sauvignon Blanc</name>
</Sauvignon_Blanc>
<Winery rdf:ID="Corton_Montrachet">
  <produces>
    <White_Burgundy rdf:ID="Corton_Montrachet_White_Burgundy">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Corton Montrachet White Burgundy</rdfs:label>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

    >STRONG</flavor>
    <maker rdf:resource="#Corton_Montrachet"/>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >FULL</body>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Corton Montrachet White Burgundy</name>
  </White_Burgundy>
</produces>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Corton Montrachet</rdfs:label>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Corton Montrachet</name>
</Winery>
<Pasta_with_light_cream_sauce rdf:ID="Pasta_with_white_clam_sauce">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pasta with white clam sauce</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pasta with white clam sauce</name>
</Pasta_with_light_cream_sauce>
<Sweet_dessert rdf:ID="Cake">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cake</name>
</Sweet_dessert>
<Pasta_with_heavy_cream_sauce rdf:ID="Fettucine_Alfredo">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Fettucine Alfredo</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Fettucine Alfredo</name>
</Pasta_with_heavy_cream_sauce>
<Wine_template rdf:ID="Chenin_Blanc">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine"/>

```

```

    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chenin Blanc</rdfs:label>
</Wine_template>
<Semillon rdf:ID="Kalin_Cellars_Semillon">
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >FULL</body>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >STRONG</flavor>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <maker>
        <Winery rdf:ID="Kalin_Cellars">
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Kalin Cellars</name>
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Kalin Cellars</rdfs:label>
            <produces rdf:resource="#Kalin_Cellars_Semillon" />
        </Winery>
    </maker>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Kalin Cellars Semillon</rdfs:label>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Kalin Cellars Semillon</name>
</Semillon>
<Shellfish rdf:ID="Mussels">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mussels</name>
</Shellfish>
<Winery rdf:ID="Sevre_Et_Maine">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sevre Et Maine</rdfs:label>
    <produces>
        <Muscadet rdf:ID="Sevre_Et_Maine_Muscadet">
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Sevre Et Maine Muscadet</name>
            <maker rdf:resource="#Sevre_Et_Maine" />
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Sevre Et Maine Muscadet</rdfs:label>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</color>

```

```

        <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >WHITE</color>
    </Muscadet>
</produces>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sevre Et Maine</name>
</Winery>
<Winery rdf:ID="Clos_De_La_Poussie">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Clos De La Poussie</rdfs:label>
    <produces>
        <Sancerre rdf:ID="Clos_De_La_Poussie_Sancerre">
            <maker rdf:resource="#Clos_De_La_Poussie"/>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</color>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</color>
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Clos De La Poussie Sancerre</rdfs:label>
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Clos De La Poussie Sancerre</name>
        </Sancerre>
    </produces>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Clos De La Poussie</name>
</Winery>
<Wine_grape rdf:ID="Pinot_Noir_grape">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pinot Noir grape</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pinot Noir grape</name>
</Wine_grape>
<Wine_grape rdf:ID="Cabernet_Franc_grape">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cabernet Franc grape</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Cabernet Franc grape</name>
</Wine_grape>
<Winery rdf:ID="Stonleigh">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Stonleigh</name>
    <produces>

```

```

<Sauvignon_Blanc rdf:ID="Stonleigh_Sauvignon_Blanc">
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DRY</sugar>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <maker rdf:resource="#Stonleigh"/>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DELICATE</flavor>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Stonleigh Sauvignon Blanc</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Stonleigh Sauvignon Blanc</rdfs:label>
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>
</Sauvignon_Blanc>
</produces>
</Winery>
<Sweet_fruit rdf:ID="Peach">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Peach</name>
</Sweet_fruit>
<Dark_meat_fowl rdf:ID="Goose">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Goose</name>
</Dark_meat_fowl>
<Wine_template rdf:ID="Pinot_Blanc">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pinot Blanc</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Wine_template rdf:ID="Chianti">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>

```

```

        <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#Red_wine"/>
</Wine_template>
<Winery rdf:ID="Longridge">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Longridge</name>
    <produces>
        <Red_Merlot rdf:ID="Longridge_Merlot">
            <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >DRY</sugar>
            <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >RED</color>
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Longridge Merlot</rdfs:label>
            <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >LIGHT</body>
            <maker rdf:resource="#Longridge"/>
            <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >MODERATE</flavor>
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Longridge Merlot</name>
        </Red_Merlot>
    </produces>
</Winery>
<Dark_meat_fowl rdf:ID="Duck">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Duck</name>
</Dark_meat_fowl>
<Wine_template rdf:ID="Sauvignon_Blanc">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#color"/>
            <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >WHITE</owl:hasValue>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#White_wine"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Sauvignon Blanc</rdfs:label>
</Wine_template>

```

```

<Winery rdf:ID="Sean_Thackrey">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Sean Thackrey</name>
  <produces>
    <Petite_Syrah rdf:ID="Sean_Thackrey_Sirius_Petite_Syrah">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Sean Thackrey Sirius Petite Syrah</rdfs:label>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >STRONG</flavor>
      <maker rdf:resource="#Sean_Thackrey"/>
      <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >FULL</body>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Sean Thackrey Sirius Petite Syrah</name>
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
    </Petite_Syrah>
  </produces>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Sean Thackrey</rdfs:label>
</Winery>
<Wine_template rdf:ID="White_Zinfandel">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >ROSDž</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Rose_wine"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >White Zinfandel</rdfs:label>
</Wine_template>
<j.0:PAL-CONSTRAINT rdf:ID="wines-current_00036">
  <j.0:PAL-STATEMENT rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >(forall ?wine
  (= &gt; (and
  (own-slot-not-null maker ?wine)
  (own-slot-not-null area ?wine))
  (= (area (maker ?wine))

```

```

(area ?wine))
)
) </j.0:PAL-STATEMENT>
  <j.0:PAL-DESCRIPTION rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >area of wine and maker are the same</j.0:PAL-DESCRIPTION>
  <j.0:PAL-NAME rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >area of wine and maker are the same</j.0:PAL-NAME>
</j.0:PAL-CONSTRAINT>
<Winery rdf:ID="Chateau_D_Ychem">
  <location rdf:resource="#Bordeaux_region"/>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau D Ychem</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau D Ychem</rdfs:label>
  <produces>
    <Sauterne rdf:ID="Chateau_D_Ychem_Sauterne">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Chateau D Ychem Sauterne</rdfs:label>
      <grape>
        <Wine_grape rdf:ID="Semillon_grape">
          <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Semillon grape</rdfs:label>
          <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Semillon grape</name>
        </Wine_grape>
      </grape>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Chateau D Ychem Sauterne</name>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >STRONG</flavor>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</color>
      <maker rdf:resource="#Chateau_D_Ychem"/>
    </Sauterne>
  </produces>
</Winery>
<Wine_template rdf:ID="Chablis">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```



```

        >WHITE</owl:hasValue>
        <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#White_Burgundy"/>
</Wine_template>
<Chianti rdf:ID="Chianti_Classico">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chianti Classico</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chianti Classico</name>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
</Chianti>
<Meal_course rdf:ID="wines-current_00033">
    <food>
        <Non-bland_fish rdf:ID="Salmon">
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Salmon</name>
        </Non-bland_fish>
    </food>
    <drink rdf:resource="#Pinot_Noir"/>
</Meal_course>
<Sweet_Reisling rdf:ID="Schloss_Volrad_Trochenbierenauslese_Riesling">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Schloss Volrad Trochenbierenauslese Riesling</name>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <maker>
        <Winery rdf:ID="Schloss_Volrad">
            <location rdf:resource="#German_region"/>
            <produces rdf:resource="#Schloss_Volrad_Trochenbierenauslese_Riesling"/>
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Schloss Volrad</rdfs:label>
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Schloss Volrad</name>
        </Winery>
    </maker>
    <grape rdf:resource="#Riesling_grape"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

>Schloss Volrad Trochenbierenauslese Riesling</rdfs:label>
<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>FULL</body>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>SWEET</sugar>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MODERATE</flavor>
</Sweet_Reisling>
<Wine_template rdf:ID="Cotes_Chalonnaise">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_Burgundy"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cotes Chalonnaise</rdfs:label>
</Wine_template>
<Wine_template rdf:ID="Cotes_d_Or">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Cotes d'Or</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_Burgundy"/>
</Wine_template>
<Winery rdf:ID="Chateau_Morgon">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Morgon</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau Morgon</rdfs:label>
  <produces>
    <Beaujolais rdf:ID="Chateau_Morgon_Beaujolais">
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

    >DELICATE</flavor>
    <maker rdf:resource="#Chateau_Morgon"/>
    <grape>
      <Wine_grape rdf:ID="Gamay_grape">
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Gamay grape</rdfs:label>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Gamay grape</name>
      </Wine_grape>
    </grape>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</color>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Morgon Beaujolais</rdfs:label>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Chateau Morgon Beaujolais</name>
    <tannin_level rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >LOW</tannin_level>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >LIGHT</body>
  </Beaujolais>
</produces>
</Winery>
<Winery rdf:ID="Page_Mill_Winery">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Page Mill Winery</rdfs:label>
  <produces>
    <Cabernet_Sauvignon rdf:ID="Page_Mill_Winery_Cabernet_Sauvignon">
      <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >MODERATE</flavor>
      <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >MEDIUM</body>
      <maker rdf:resource="#Page_Mill_Winery"/>
      <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >DRY</sugar>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Page Mill Winery Cabernet Sauvignon</name>
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Page Mill Winery Cabernet Sauvignon</rdfs:label>
    </Cabernet_Sauvignon>
  </produces>
</Winery>

```

```

    </produces>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Page Mill Winery</name>
</Winery>
<Non-bland_fish rdf:ID="Tuna">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Tuna</name>
</Non-bland_fish>
<Bland_fish rdf:ID="Flounder">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Flounder</name>
</Bland_fish>
<Chardonnay rdf:ID="Peter_Mccoy_Chardonnay">
    <maker>
        <Winery rdf:ID="Peter_Mccoy">
            <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Peter Mccoy</name>
            <produces rdf:resource="#Peter_Mccoy_Chardonnay" />
            <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Peter Mccoy</rdfs:label>
        </Winery>
    </maker>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MODERATE</flavor>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Peter Mccoy Chardonnay</name>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Peter Mccoy Chardonnay</rdfs:label>
    <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >WHITE</color>
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DRY</sugar>
    <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >MEDIUM</body>
</Chardonnay>
<Dessert_wine rdf:ID="Whitehall_Lane_Primavera">
    <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >SWEET</sugar>
    <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >DELICATE</flavor>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Whitehall Lane Primavera</rdfs:label>

```

```

<body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>LIGHT</body>
<maker>
  <Winery rdf:ID="Whitehall_Lane">
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Whitehall Lane</name>
    <produces>
      <Cabernet_Franc rdf:ID="Whitehall_Lane_Cabernet_Franc">
        <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >DRY</sugar>
        <maker rdf:resource="#Whitehall_Lane"/>
        <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >MODERATE</flavor>
        <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Whitehall Lane Cabernet Franc</name>
        <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >MEDIUM</body>
        <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >RED</color>
        <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Whitehall Lane Cabernet Franc</rdfs:label>
      </Cabernet_Franc>
    </produces>
    <produces rdf:resource="#Whitehall_Lane_Primavera"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Whitehall Lane</rdfs:label>
  </Winery>
</maker>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Whitehall Lane Primavera</name>
</Dessert_wine>
<White_Burgundy rdf:ID="Chateau_De_Meursault_Meursault">
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MODERATE</flavor>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau De Meursault Meursault</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Chateau De Meursault Meursault</name>
</White_Burgundy>
<Wine_template rdf:ID="Cabernet_Sauvignon">

```

```

<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Cabernet Sauvignon</rdfs:label>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#color"/>
    <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >RED</owl:hasValue>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#Red_wine"/>
</Wine_template>
<Wine_grape rdf:ID="Sauvignon_Blanc_grape">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Sauvignon Blanc grape</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Sauvignon Blanc grape</name>
</Wine_grape>
<Wine_grape rdf:ID="Petite_Verdot_grape">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Petite Verdot grape</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Petite Verdot grape</rdfs:label>
</Wine_grape>
<Tomato-based_food rdf:ID="Pizza">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pizza</name>
</Tomato-based_food>
<Wine_template rdf:ID="Graves">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_Bordeaux"/>
</Wine_template>
<Wine_grape rdf:ID="Zinfandel_grape">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Zinfandel grape</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Zinfandel grape</rdfs:label>

```

```

</Wine_grape>
<Shellfish rdf:ID="Lobster">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Lobster</name>
</Shellfish>
<Wine_template rdf:ID="Margaux">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Medoc"/>
</Wine_template>
<Winery rdf:ID="Clos_De_Vougeot">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Clos De Vougeot</rdfs:label>
  <produces>
    <Red_Burgundy rdf:ID="Clos_De_Vougeot_Cotes_D_Or">
      <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Clos De Vougeot Cotes D Or</rdfs:label>
      <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</color>
      <maker rdf:resource="#Clos_De_Vougeot"/>
      <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Clos De Vougeot Cotes D Or</name>
    </Red_Burgundy>
  </produces>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Clos De Vougeot</name>
</Winery>
<Non-spicy_red_meat rdf:ID="Veal">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Veal</name>
</Non-spicy_red_meat>
<Winery rdf:ID="Lane_Tanner">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Lane Tanner</rdfs:label>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Lane Tanner</name>
  <produces>

```

```

<Pinot_Noir rdf:ID="Lane_Tanner_Pinot_Noir">
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >LIGHT</body>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Lane Tanner Pinot Noir</rdfs:label>
  <flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DELICATE</flavor>
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >RED</color>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Lane Tanner Pinot Noir</name>
  <sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >DRY</sugar>
  <maker rdf:resource="#Lane_Tanner"/>
</Pinot_Noir>
</produces>
</Winery>
<Meal_course rdf:ID="wines-current_00031">
  <drink rdf:resource="#Chablis"/>
  <food rdf:resource="#Oysters"/>
</Meal_course>
<Wine_grape rdf:ID="Malbec_grape">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Malbec grape</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Malbec grape</rdfs:label>
</Wine_grape>
<Shellfish rdf:ID="Crabs">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Crabs</name>
</Shellfish>
<Pasta_with_non-spicy_red_sauce rdf:ID="Spaghetti_with_tomato_sauce">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Spaghetti with tomato sauce</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Spaghetti with tomato sauce</rdfs:label>
</Pasta_with_non-spicy_red_sauce>
<Semillon rdf:ID="Congress_Springs_Semillon">
  <color rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WHITE</color>
  <body rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >MEDIUM</body>

```



```

<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Congress Springs Semillon</rdfs:label>
<sugar rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>DRY</sugar>
<maker>
  <Winery rdf:ID="Congress_Springs">
    <produces rdf:resource="#Congress_Springs_Semillon" />
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Congress Springs</name>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Congress Springs</rdfs:label>
  </Winery>
</maker>
<flavor rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>MODERATE</flavor>
<name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Congress Springs Semillon</name>
</Semillon>
<Wine_template rdf:ID="Semillon">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine" />
</Wine_template>
<White_meat rdf:ID="Pork">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Pork</name>
</White_meat>
<Wine_template rdf:ID="Red_Merlot">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color" />
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Red_wine" />
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

    >Red Merlot</rdfs:label>
</Wine_template>
<Non-spicy_red_meat rdf:ID="Roast_beef">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Roast beef</name>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Roast beef</rdfs:label>
</Non-spicy_red_meat>
<Wine_template rdf:ID="Sancerre">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >RED</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#color"/>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#White_wine"/>
</Wine_template>
<Wine_template rdf:ID="Ice_Wine">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Ice Wine</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >WHITE</owl:hasValue>
      <owl:onProperty rdf:resource="#color"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >SWEET</owl:hasValue>
      <owl:onProperty rdf:resource="#sugar"/>
    </owl:Restriction>
  </rdfs:subClassOf>

```

```
<rdfs:subClassOf rdf:resource="#White_wine"/>
<rdfs:subClassOf rdf:resource="#Dessert_wine"/>
</Wine_template>
</rdf:RDF>
```

```
<!-- Created with Protege (with OWL Plugin 3.3.1, Build 430) http://protege.stanford.edu -->
```