



DERI INNSBRUCK

Leopold Franzens
Universität Innsbruck



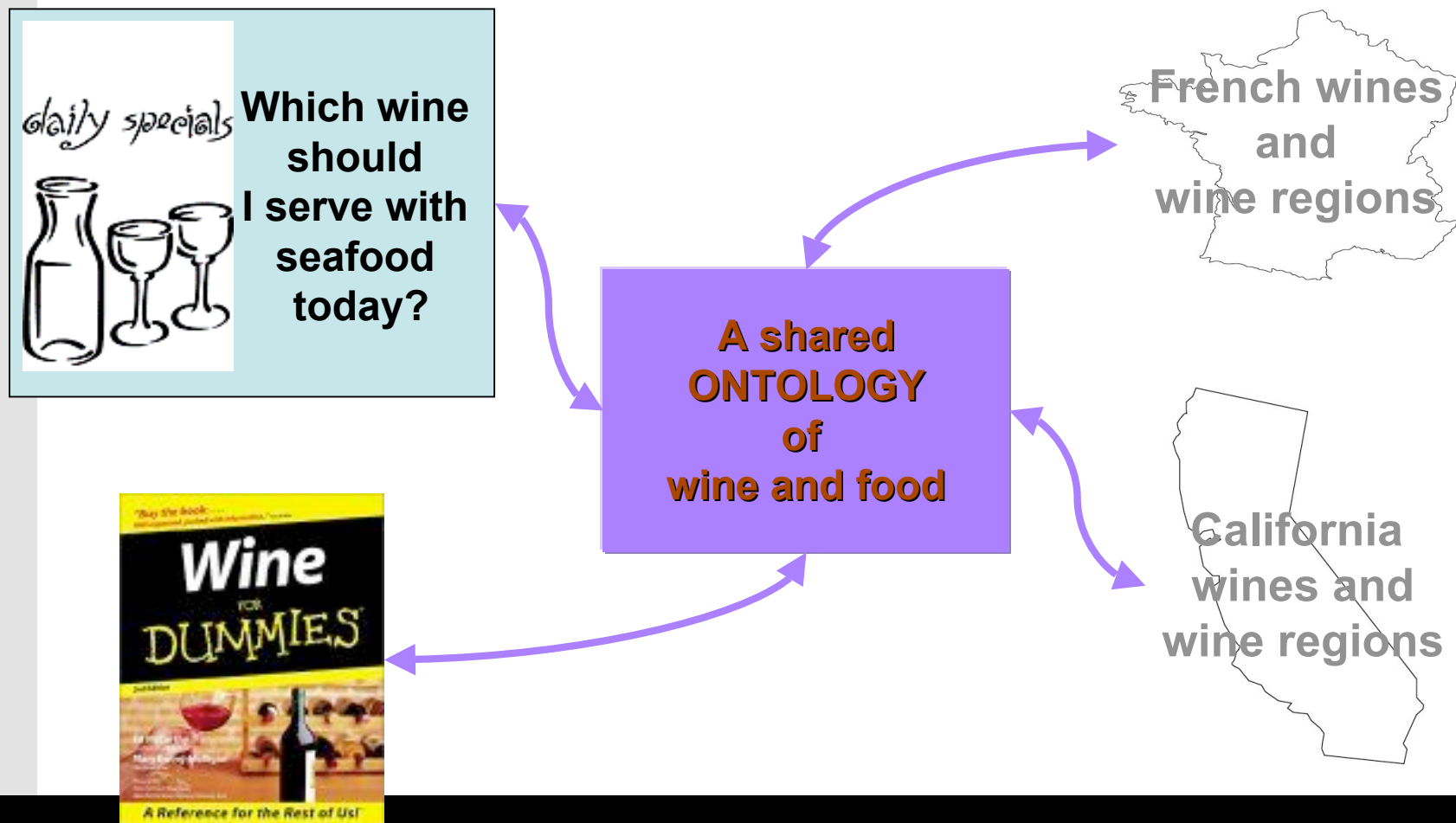
Tutorial Ontology Modeling*

Elena Simperl, University of Innsbruck

**[Natalya F. Noy](#) and [Deborah L. McGuinness](#). "Ontology Development 101: A Guide to Creating Your First Ontology". Stanford Knowledge Systems Laboratory Technical Report [KSL-01-05](#) and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.*



<http://protege.cim3.net/file/pub/ontologies/wine/wine.owl>





Step 1: Determine the domain and scope of the ontology

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?



- A set of queries which place demands on the underlying ontology.
- Ontology must be able to represent the questions using its terminology and the answers based on the axioms
- Ideally, in a staged manner, where consequent questions require the input from the preceeding ones.
- A rationale for each competency question should be given.



Example: Competency Questions

- Which wine characteristics should I consider when choosing a wine?
- Is Bordeaux a red or white wine?
- Does Cabernet Sauvignon go well with seafood?
- What is the best choice of wine for grilled meat?
- Which characteristics of a wine affect its appropriateness for a dish?
- Does a flavor or body of a specific wine change with vintage year?
- What were good vintages for Napa Zinfandel?



Step 2: Consider reusing existing ontologies

- Reuse ensures interoperability and reduces costs
- Ontology libraries and tools for customization are required for this step
- You can also reuse lightweight knowledge structures such as taxonomies or even perform ontology learning...
- Sub-steps
 - Discover potential reuse candidates
 - Evaluate their usability
 - Customize ontologies to be reused
 - Integrate and merge to the target ontology



Step 3: Enumerate important terms in the ontology

- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?



Enumerating Terms - The Wine Ontology

*wine, grape, winery, location, wine color, wine body,
wine flavor, sugar content, white wine, red wine,
Bordeaux wine, food, seafood, fish, meat, vegetables,
cheese...*

and not

*sightseeing Tuscany, atoms and molecules of
alcohol, underage drinking laws...*



Step 4: Define the classes and the class hierarchy

- A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.
- A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.
- Middle-out approach: define the more salient concepts first and then generalize and specialize them appropriately.



Step 4: Define the classes and the class hierarchy (ii)

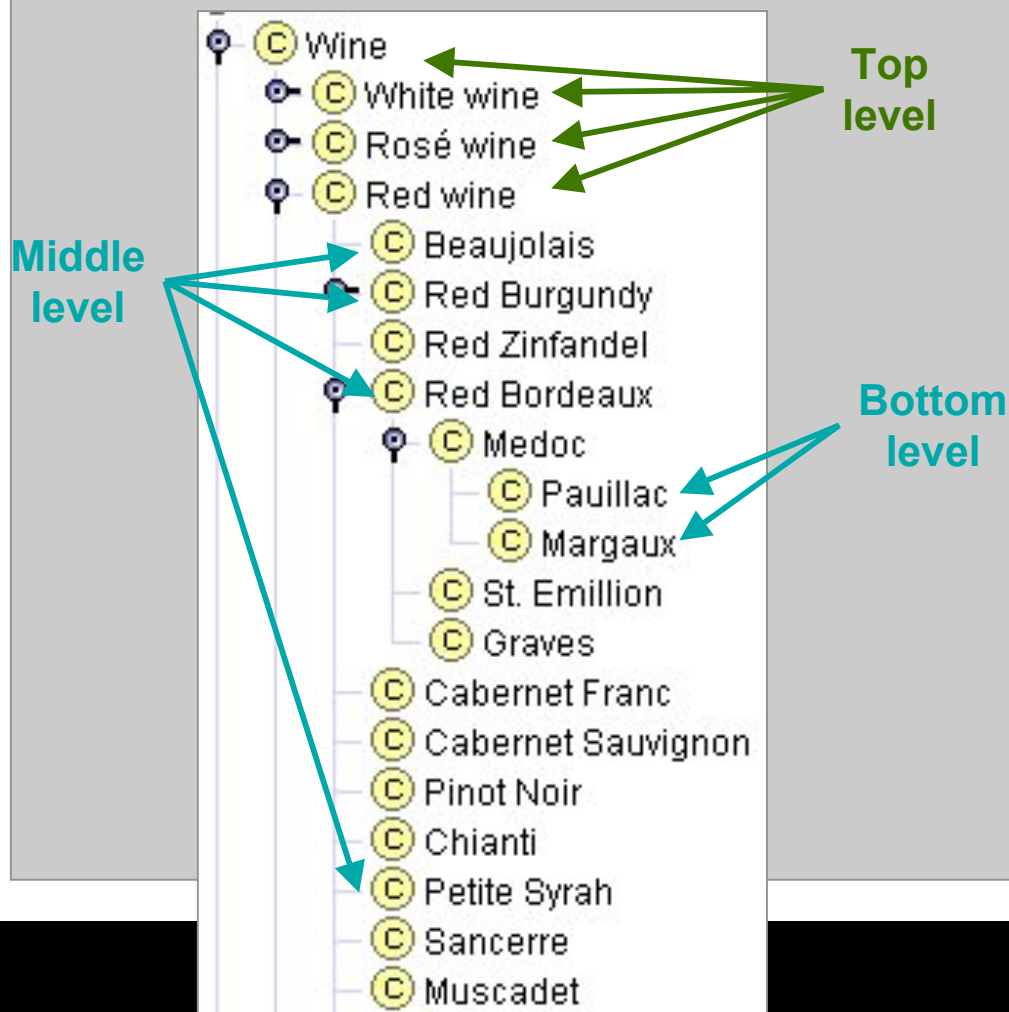
- From the list created in Step 3, select the terms that describe objects having independent existence rather than terms that describe these objects.
 - These terms will be classes in the ontology
- Organize the classes into a hierarchical taxonomy by asking if by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other class.
 - *If a class A is a superclass of class B, then every instance of B is also an instance of A*
- Classes as unary predicates—questions that have one argument. For example, “Is this object a wine?”
 - By contrast binary predicates (or slots)—questions that have two arguments. For example, “Is the flavor of this wine strong?”
“What is the flavor of this wine?”



- Apple is a subclass of Fruit
Every apple is a fruit
- Red wines is a subclass of Wine
Every red wine is a wine
- Chianti wine is a subclass of Red wine
Every Chianti wine is a red wine



Levels in the Hierarchy





- Classes (and their properties) usually have documentation
 - Describing the class in natural language
 - Listing domain assumptions relevant to the class definition
 - Listing synonyms
- Documenting ontologies is as important as documenting computer code!



Step 5: Define the properties of classes

- Step 4 selected classes from the list of terms we created in Step 3.
 - Most of the remaining terms are likely to be properties of these classes.
 - For each property in the list, we must determine which class it describes
- *Each wine will have color, sugar content, producer, etc.*
- Types of properties
 - “intrinsic” properties: **flavor and color of wine**
 - “extrinsic” properties: **price and producer of wine**
 - parts, if the object is structured (physical or abstract): **sub-regions of a regions**
 - relationships to other individuals: **wine isProducedBy producer**



- Properties are inherited and should be attached to the most general class in the hierarchy
- A subclass inherits all the properties from the superclass

If a wine has a name and flavor, a red wine also has a name and flavor

- If a class has multiple superclasses, it inherits properties from all of them

Port is both a dessert wine and a red wine. It inherits “sugar content: high” from the former and “color:red” from the latter



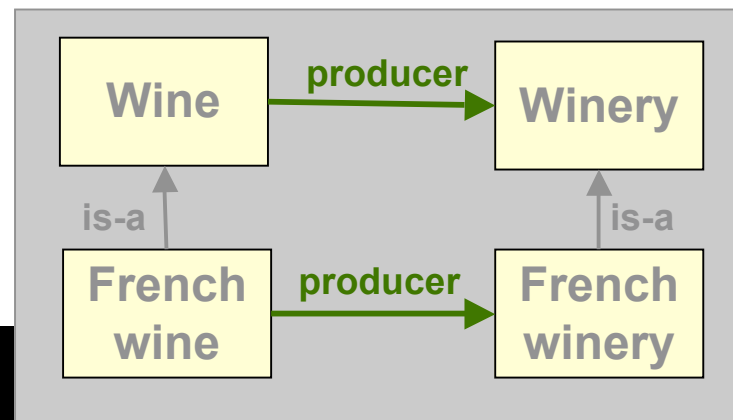
Step 6: Define the restrictions of the properties

- Refine the semantics of the properties
 - Cardinality
 - Domain and range
 - When defining a domain or a range for a slot, find the most general classes or class that can be respectively the domain or the range for the slots .
 - Do not define a domain and range that is overly general
- *The name of a wine is a string*
- *The wine producer is an instance of Winery*
- *A winery has exactly one location*



- Cardinality – the number of values a property has
- Value type – the type of values a property has
- Minimum and maximum value – a range of values for a numeric property
- Default value – the value a property has unless explicitly specified otherwise

- A subclass inherits all the properties from the superclass
- A subclass can override the constraints to “narrow” the list of allowed values
 - Make the cardinality range smaller
 - Replace a class in the range with a subclass





- Define an individual instance of a class requires
 - choose a class
 - create an individual instance of that class
 - filling in the values of the properties



DERI INNSBRUCK

Leopold Franzens
Universität Innsbruck



Practical modeling guidelines



- There is no one correct way to model a domain— there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.
- Ontology development is necessarily an iterative process.
- Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.



- The more formal an ontology is, the more does it exclude unwanted interpretations and includes the amount of inferences that can be drawn.
- However, it is more costly in terms of labor and engineering delay to create a more formal conceptualization.
- Also, achieving consensus is more difficult and time-consuming for a higher degree of formality.
- A higher degree of formality also imposes high entry barriers on the ones who are to model the respective domains and thus excludes potential contributors.



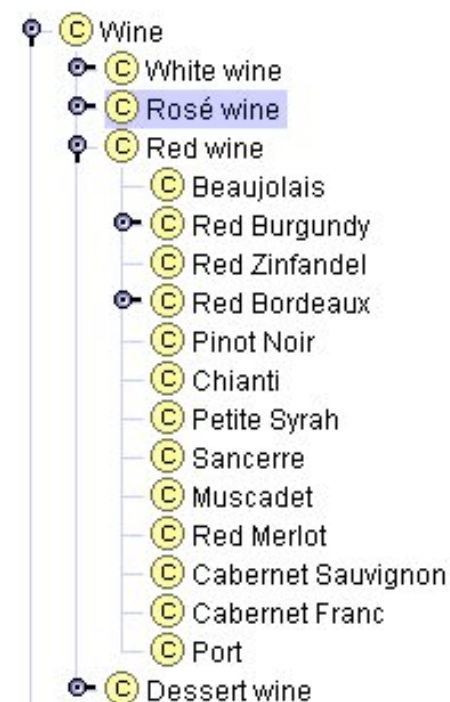
- Be clear about why the ontology is being built and what its intended usages are
 - Interoperability between systems
 - Systems engineering
 - Semantic search, semantic annotation
 - Communication between people and organizations
- Example: semantic search
 - Semi-formal ontology
 - Usage of natural language labels and naming conventions
 - Well-balanced at schema and instance level
 - Rich conceptualization
 - Syntactical and semantic correctness
- The ontology should not contain all the possible information about the domain



- A subclass of a class represents a concept that is a “kind of” the concept that the superclass represents.
- Classes represent concepts in the domain and not the words that denote these concepts.
- *A single person is not a subclass of all persons*
- Synonyms for the same concept do not represent different classes



- All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality.
- If a class has only one direct subclass there may be a modeling problem or the ontology is not complete.
- If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary.





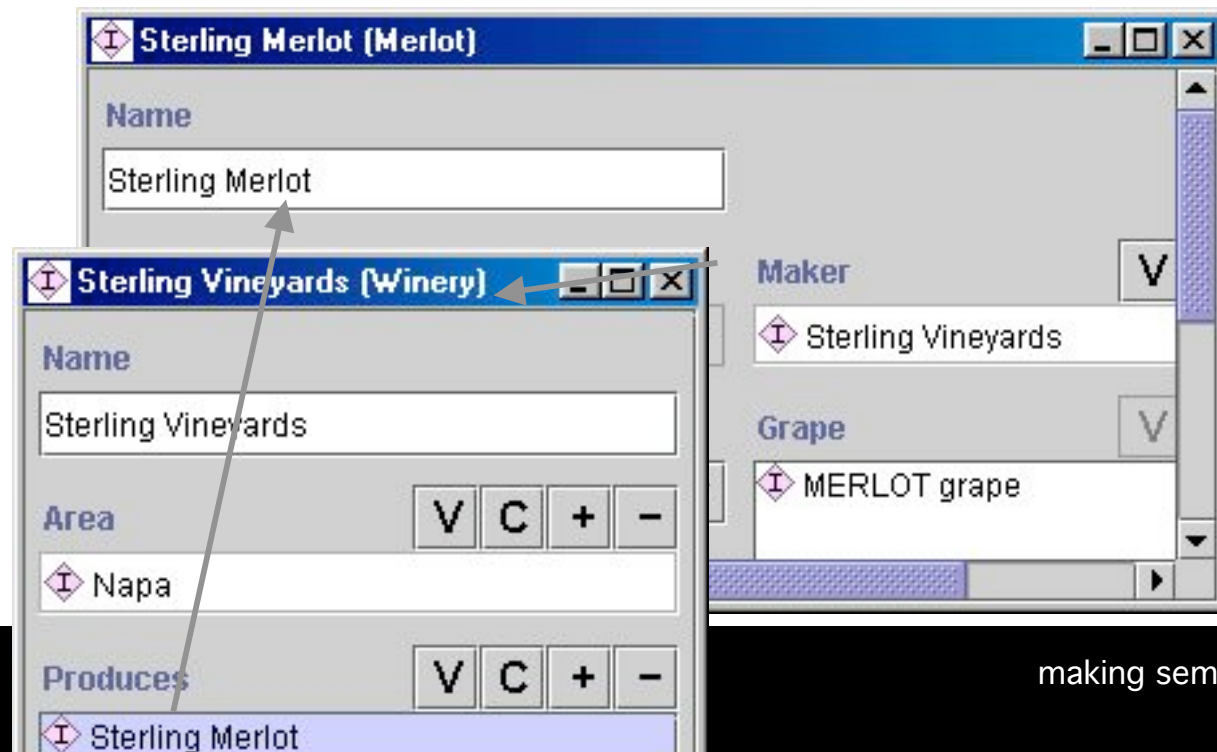
- Subclasses of a class usually
 - have additional properties that the superclass does not have, or
 - restrictions different from those of the superclass, or
 - participate in different relationships than the superclasses



Domain and range of properties

- When defining a domain or range find the most general class or classes
- *The domain of flavor should be Wine and not Red wine or White wine*
- *The range of produces for a Winery should be Wine and not Red, White or Rosé wine*
- General patterns
 - A class and a superclass – replace with the superclass
 - All subclasses of a class – replace with the superclass
 - Most subclasses of a class – consider replacing with the superclass

- Modeling with inverse properties is redundant, but
 - Allows acquisition of the information in either direction
 - Enables additional verification
 - Allows presentation of information in both directions
- The actual implementation differs from system to system
 - Are both values stored?
 - When are the inverse values filled in?



making semantic real.



DERI INNSBRUCK

Leopold Franzens
Universität Innsbruck



Thank you for your attention