MANCHESTER 1824

The University of Manchester

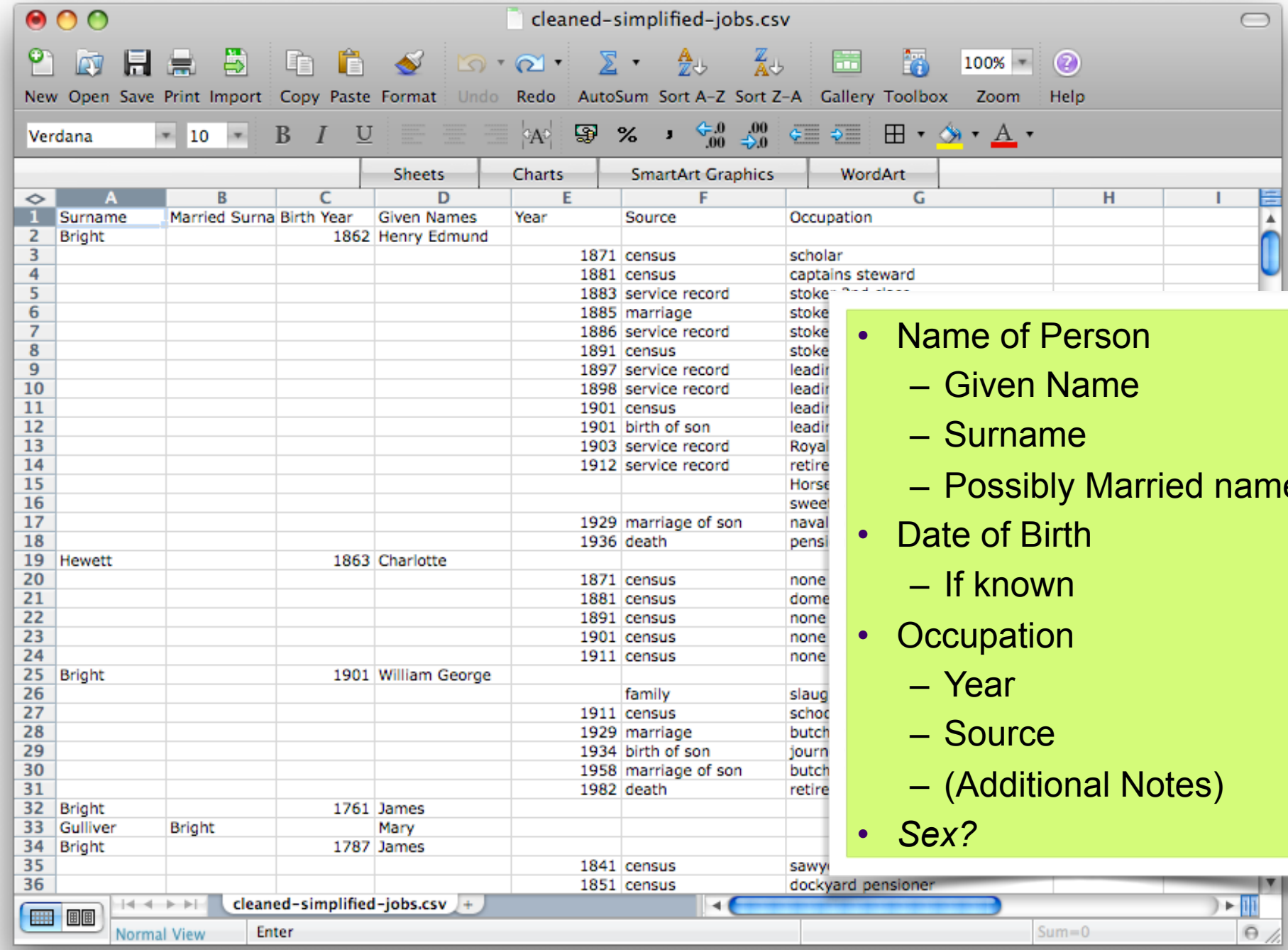
# Modelling Roles

COMP62342
Sean Bechhofer
sean.bechhofer@manchester.ac.uk

*With thanks to Robert Stevens*

# Occupations

- Imagine we have some information about people and their occupations (and where that information is derived from)
- How might we define an ontology to allow us to describe this data?
- What are the classes we need to represent?
- What are the properties or relationships that we need to describe?
- How can we map from spreadsheet data into some populated ontology?
- What queries can we then ask?




- Name of Person
  - Given Name
  - Surname
  - Possibly Married name
- Date of Birth
  - If known
- Occupation
  - Year
  - Source
  - (Additional Notes)
- *Sex?*

# Modelling in OWL

- Recall that OWL allows us to describe
  - Individuals.
  - Classes (of Individuals).
  - Relationships between Individuals or Properties of Individuals.

- What are our Individuals here?
- What are the Classes
- What are the Properties/Relations?

# Basic Data

- Each Person has
  - Given Name
  - Surname
  - Date of Birth
- Some Persons (Women) may also have
  - Married Surname

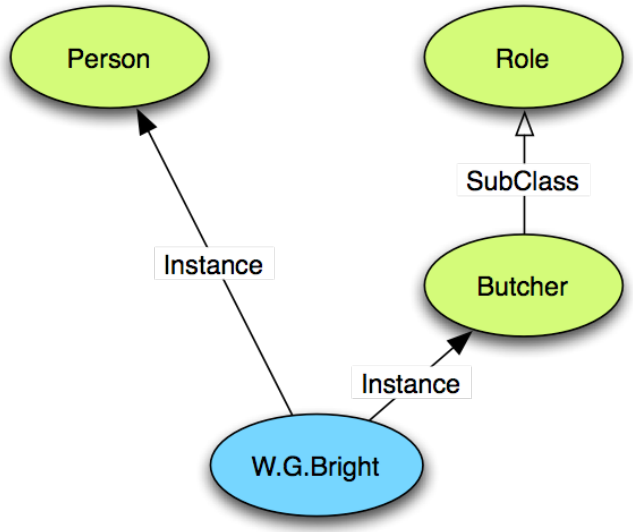
- OWL provides *Datatype* properties that allow us to associate data values with Individuals.
  - Strings, numbers etc.

# Occupations

- We are assuming that we have a hierarchy of occupations or roles (not all of the things that people are listed as doing are necessarily occupations)
- This is a simple taxonomy.
- We might, at some point, be concerned about modelling this more completely, e.g. through descriptions of the roles, but for current purposes, an asserted hierarchy is fine.

- However, a key question is how we associate people with the occupations/roles that they are playing.

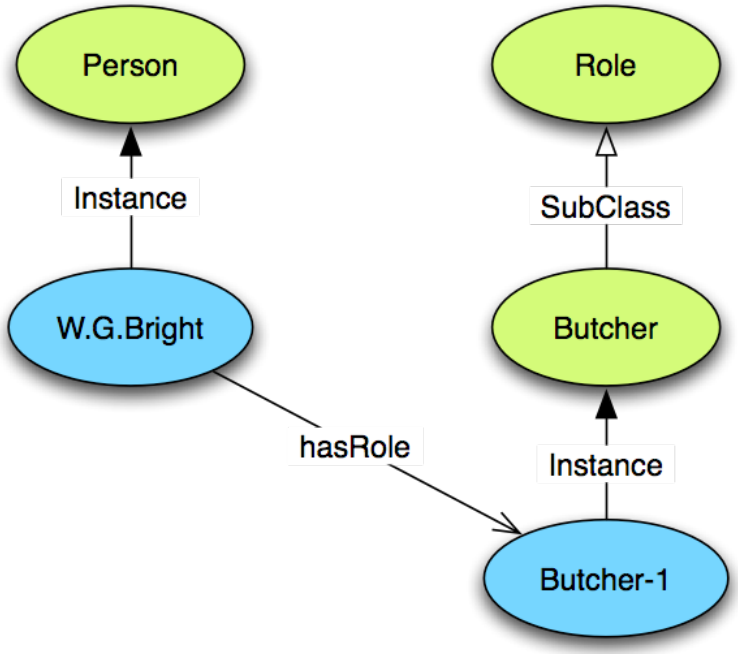
# Modelling Occupations: Attempt #1




Class: Person
Class: Role
Class: Butcher
SubClassOf: Role

Individual: W.G.Bright
Types: Person, Butcher

# Modelling Occupations: Attempt #2




Class: Person
Class: Role
Class: Butcher
SubClassOf: Role

ObjectProperty: hasRole

Individual: Butcher-1
Types: Butcher
Individual: W.G.Bright
Types: Person
Facts: hasRole Butcher-1

# Named and Anonymous Individuals

- OWL allows us to make statements about particular named individuals.
- Fred has a cat called Tibbs.

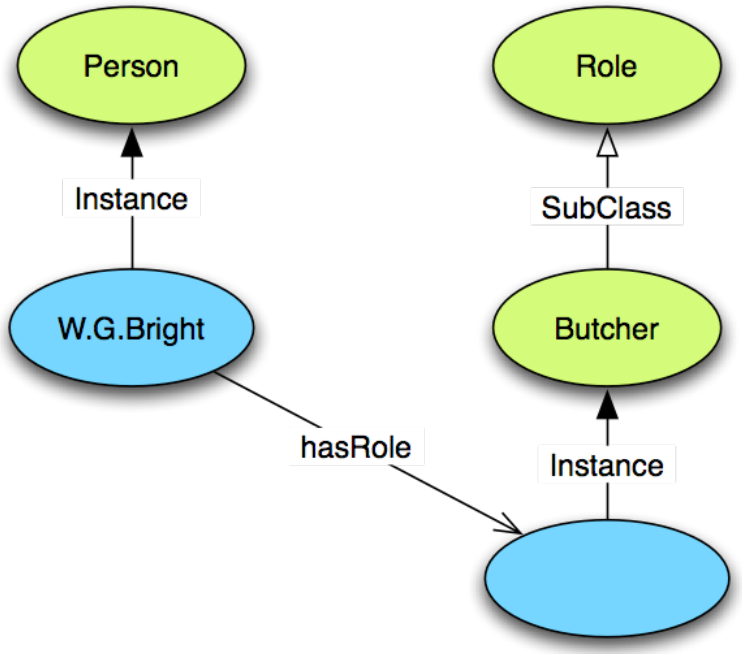
Individual: Fred
Types: Person
Facts: hasPet Tibbs

Individual: Tibbs
Types: Cat

- We can also refer to *anonymous* individuals
- Fred has a cat, but we don't know anything about it
- This representation of *incomplete* information can be useful when we don't know (or don't care) about the particular individual.

Individual: Fred
Types: Person that hasPet some Cat

# Modelling Occupations: Attempt #3




Class: Person

Class: Role

Class: Butcher

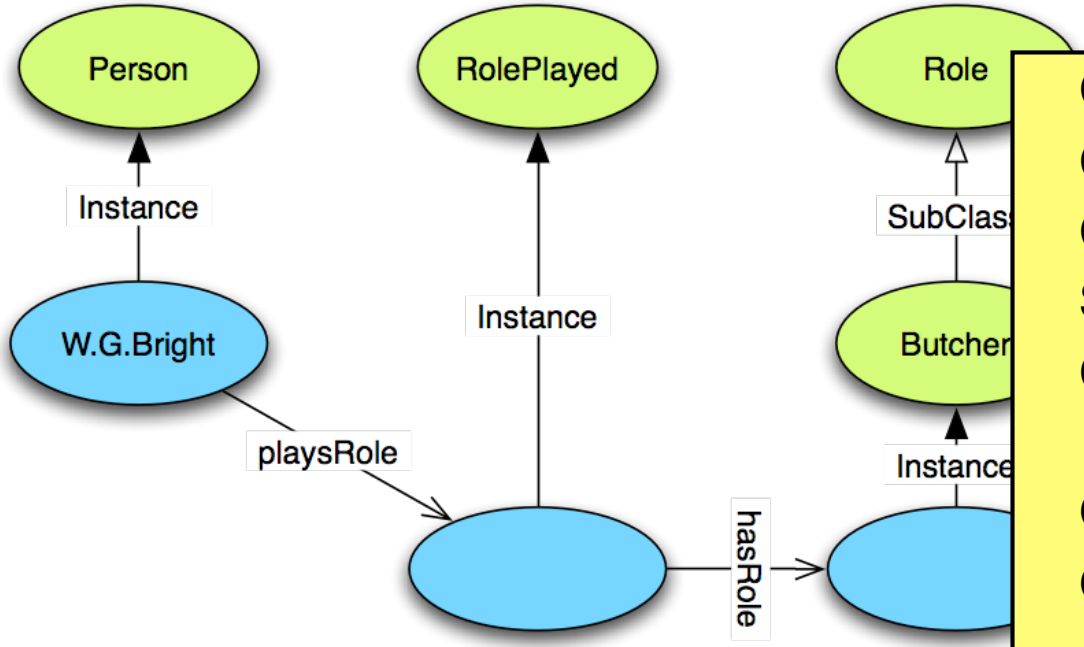
SubClassOf: Role

ObjectProperty: hasRole

Individual: W.G.Bright

Types: Person that hasRole some Butcher

# Modelling Occupations: Attempt #4




Class: Person

Class: Role

Class: Butcher

SubClassOf: Role

Class: RolePlayed

ObjectProperty: playsRole

ObjectProperty: hasRole

Individual: W.G.Bright

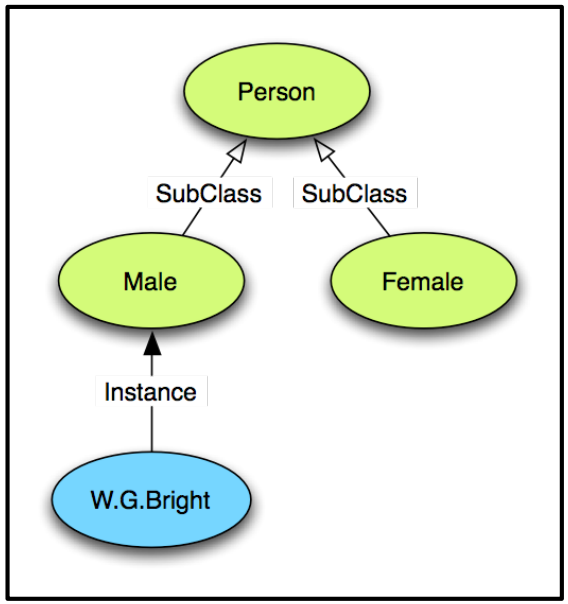
Types: Person that playsRole some (RolePlayed that hasRole some Butcher)

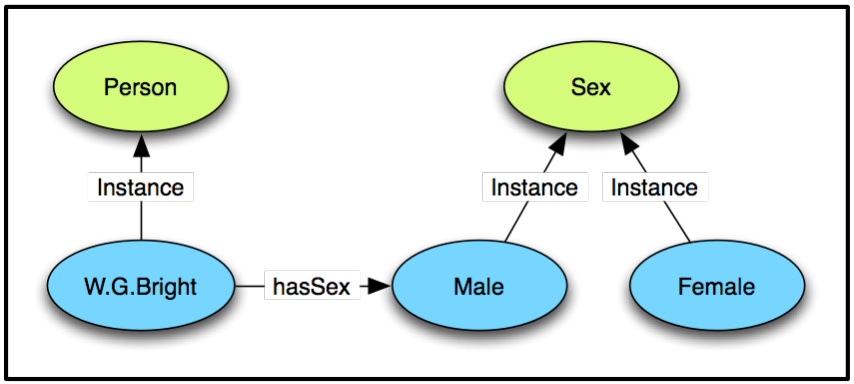
# Modelling Sex: Male and Female

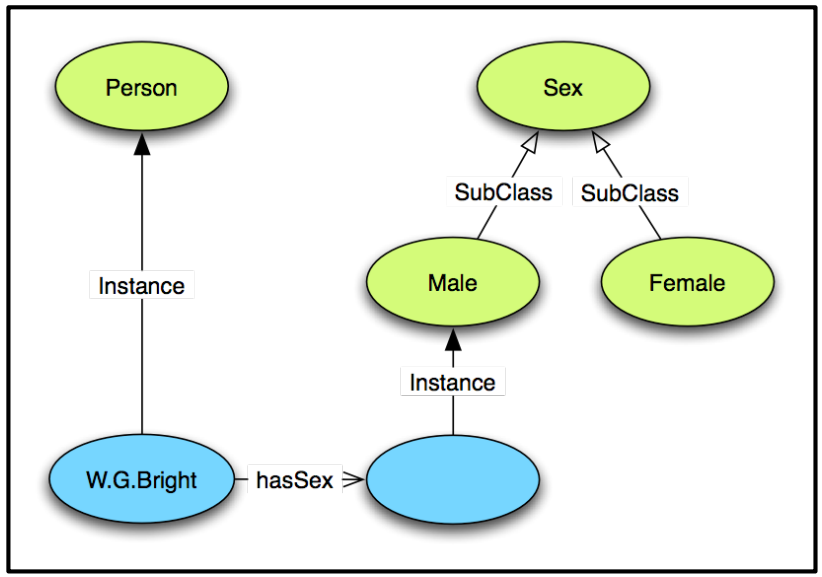
- People are Male or Female
  - For the purposes of this model we will take a simplified view and not consider transgender/androgeny etc.
  - Thus every person is either Male or Female and not both

A. Subclasses of Person Male, Female, with Individuals being asserted as instances of those classes
   - Disjointness and Covering Axioms
B. Two distinct Individuals Male, Female with a functional ObjectProperty hasSex
C. Classes Male, Female, a functional ObjectProperty hasSex with Individuals being asserted to be related to anonymous Individuals of those classes.
   - Disjointness and Covering Axioms

# Modelling Male and Female




- Incomplete information?
- Extensibility?
- Shared “maleness”?
- Definitions of Man and Woman?

- How can we tell if people are Male or Female in the data?

# Modelling Dates

- The data states a date for the occupation
  - Start date?
  - End date?
  - Some kind of duration?

- A simple approach here is to provide a DatatypeProperty that associates the year (as an integer) with the RolePlayed instance
  - An advantage of considering the RolePlayed as an object.
  - Other data types are available. e.g. for dates

- Limitations?

# Provenance

- It is often important to maintain the *provenance* of information.
  - Where does this fact come from?
- The data includes this for most of the facts
  - Census Records
  - Birth Records
  - Death Records
  - Marriage Records
- We can provide additional attributes on the RolePlayed Individuals stating where the information came from

- As with Sex there are possible choices
  - Distinct Individuals: Census, BirthRecord, DeathRecord etc.
  - Classes for Source types
  - Classes for Source types with Anonymous Individuals.
  - Would we ever want to name these?

# Named or Anonymous RolePlayed Object?

```
Individual: William_George_Bright_1901
Types: playsRole some (RolePlayed
                and hasRole some Butcher
                and hasYear value "1929"^^integer
                and hasSource some MarriageRecord)
```

```
Individual: James_Bright_1809
Facts: playsRole rolePlayed_001

Individual: rolePlayed_001
Types: RolePlayed and (hasRole some Seaman)
Facts: hasYear “1839”^^integer
```