Modelling : some basic principles

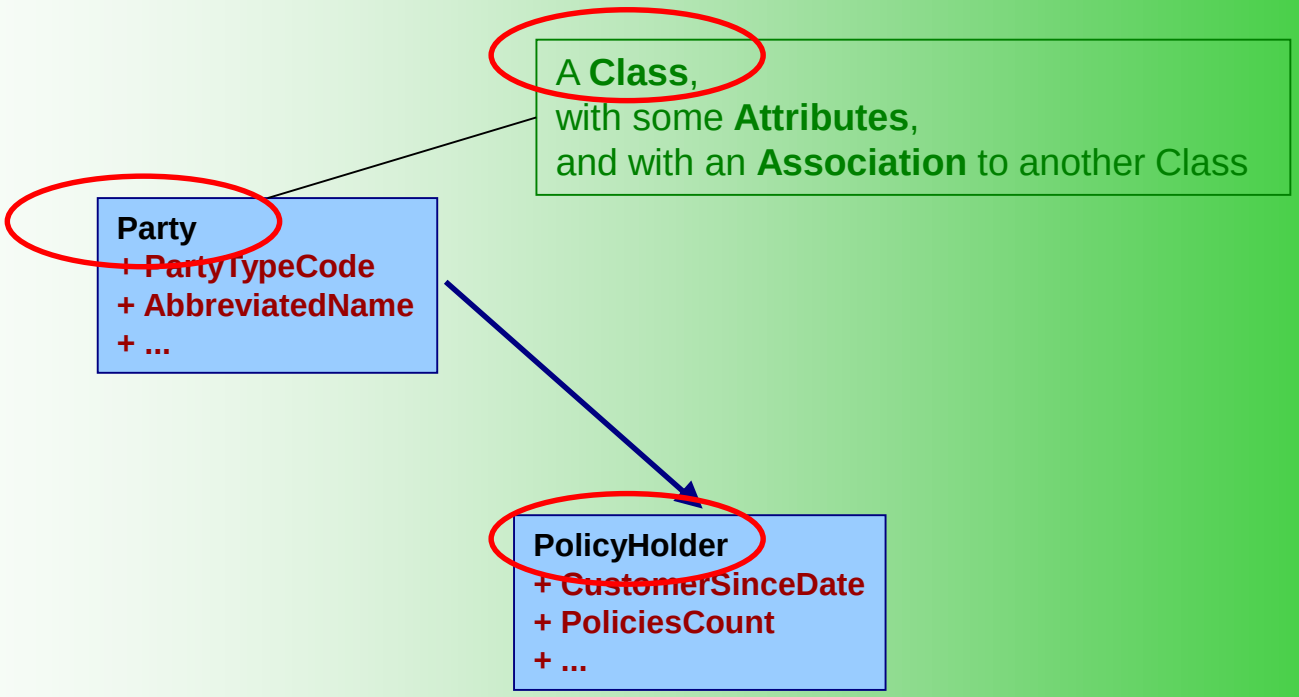
- Classes
- Attributes of classes
- Associations from/to classes
- Generalizations of classes
- Extensions of classes
- Inheritance

Modelling : some basic principles

- **Classes**
  - The purpose of a class is to specify **a classification of objects** and to specify **the features** that characterize the structure and behavior **of those objects**.
  - Objects of a class must contain values for each attribute that is a member of that class, in accordance with the characteristics of the attribute, for example its type and multiplicity.

(As stated by the OMG UML specification (UML Superstructure Specification, v2.1.1, pp. 52-53).)

Modelling : some basic principles



(*A Party can play the role of PolicyHolder.*)

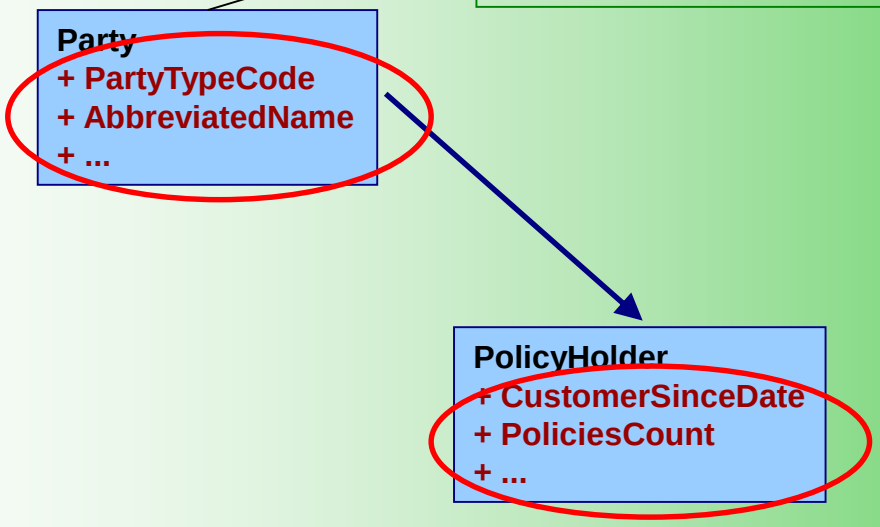
Modelling : some basic principles

- **Attributes of classes**
  - Attributes are features of a Class or other element that represent the properties or internal data elements of that element.

    For a Customer Class, CustomerName and CustomerAddress can be attributes.

Modelling : some basic principles

A **Class**,
with some **Attributes**,
and with an **Association** to another Class



(*A Party can play the role of PolicyHolder.*)

Modelling : some basic principles

- **Attributes of classes**

  (As stated in some other context :
  "**Value and Entity Types**"

  - A *value type* is a type for something which can have an utterable value, such as words and numbers. For example the name of a person is a value type.
  - The person is an *entity* because when we point to a person we have no utterable value for it.
  - His age, name and weight are again utterable values
)

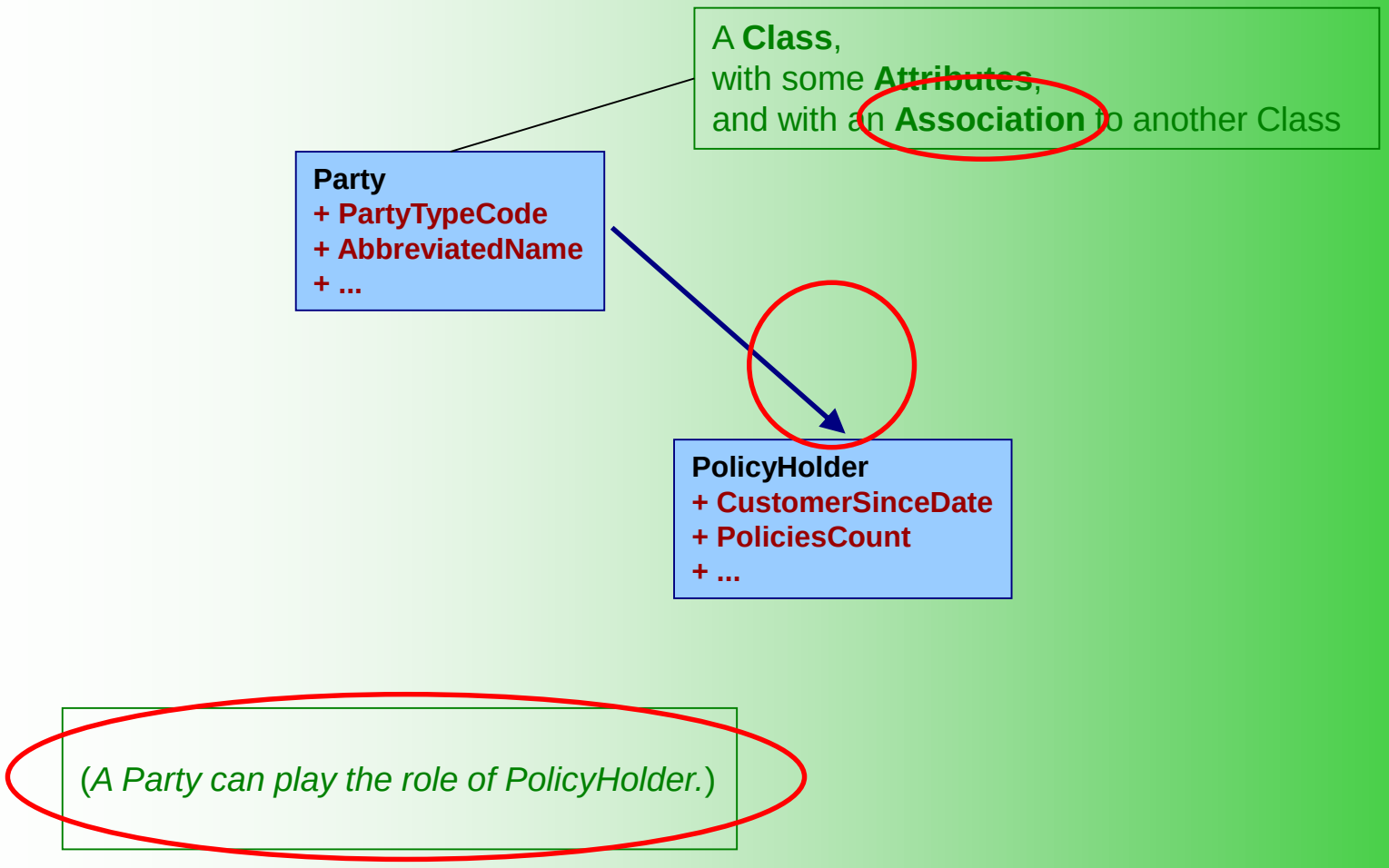
Modelling : some basic principles

- **Associations from/to classes**
  - An association specifies a semantic relationship that can occur between typed instances.
    It has at least two ends represented by properties, each of which is connected to the type of the end.
    More than one end of the association may have the same type.

  - An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.

  (As stated by the OMG UML specification (UML Superstructure Specification, v2.1.1, p. 41).)

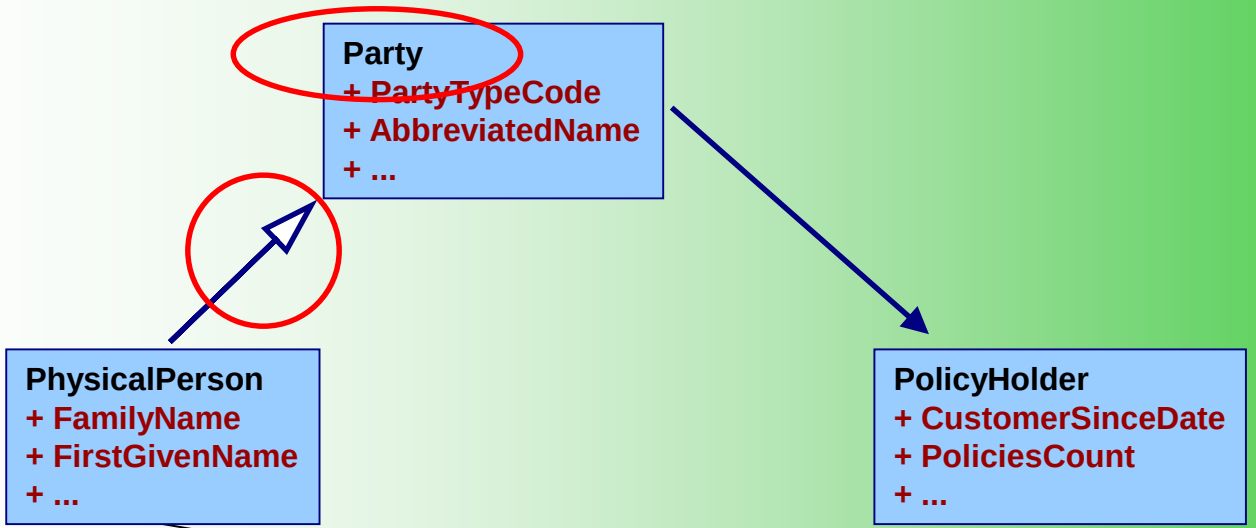
Modelling : some basic principles



Modelling : some basic principles

- **Generalizations of classes**
  - A generalization is a taxonomic relationship between a more general classifier and a more specific classifier.

Modelling : some basic principles



- **Party** is the **Generalization** of PhysicalPerson
- **PhysicalPerson** is the **Extension** of Party
- **PhysicalPerson inherits**
  - the Attributes of Party
  - the Associations of Party

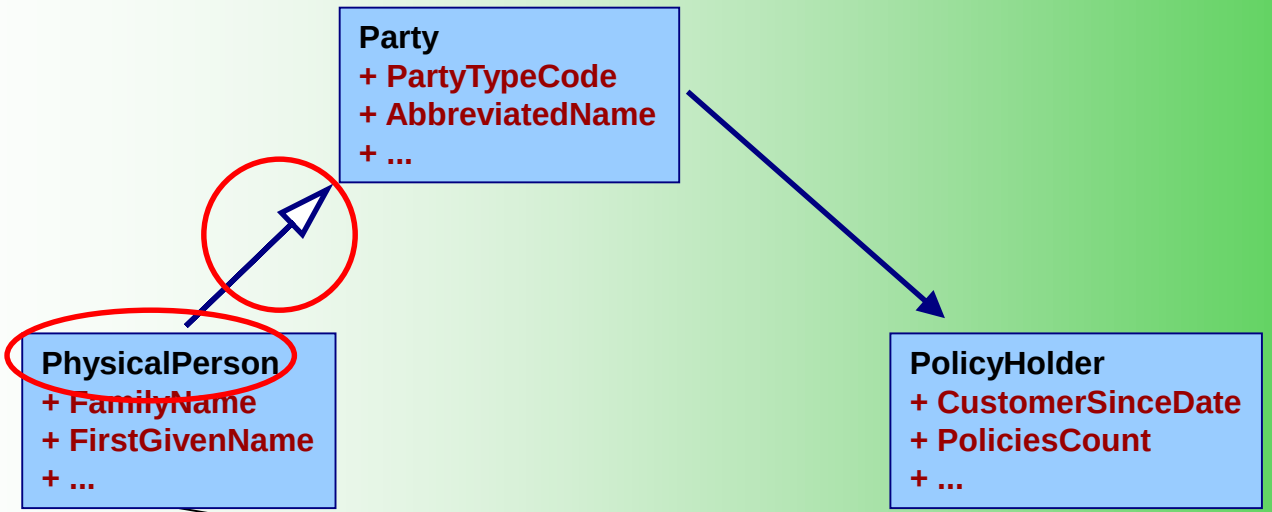
(A PhysicalPerson is a Party, but not every Party is a PhysicalPerson.
Party does not inherit from PhysicalPerson.)

(***A PhysicalPerson is a Party, and so can play the role of PolicyHolder.***)

Modelling : some basic principles

- **Extensions of classes**
  - A generalization is a taxonomic relationship between a more general classifier and a more specific classifier.
  - **Each instance of the specific classifier is also an indirect instance of the general classifier.**

Modelling : some basic principles



- **Party** is the **Generalization** of PhysicalPerson
- **PhysicalPerson** is the **Extension** of Party
- **PhysicalPerson inherits**
  - the Attributes of Party
  - the Associations of Party

(A PhysicalPerson is a Party, but not every Party is a PhysicalPerson.
Party does not inherit from PhysicalPerson.)

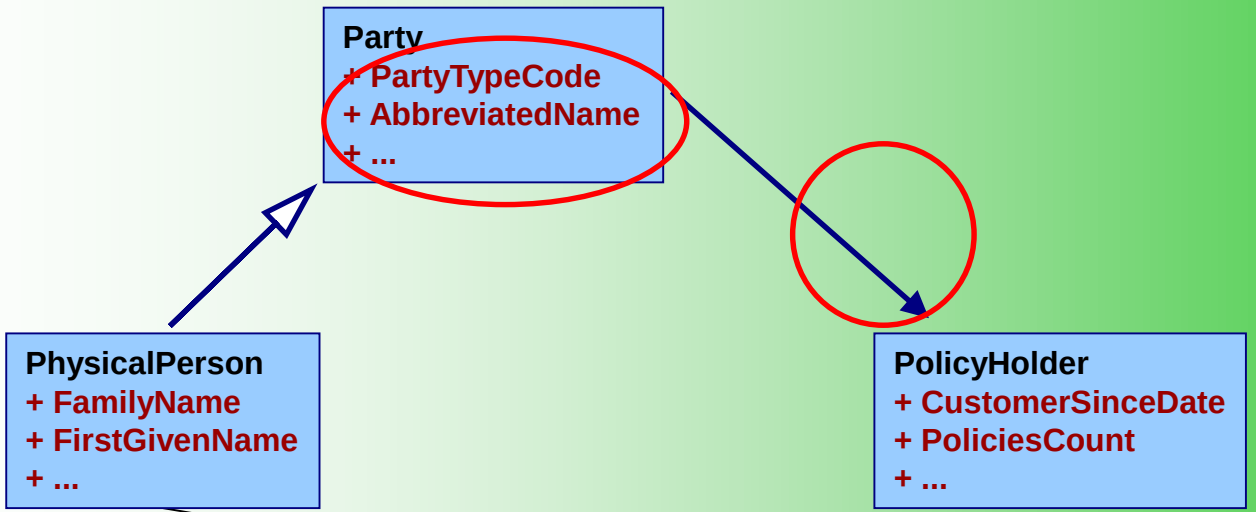
(***A PhysicalPerson is a Party, and so can play the role of PolicyHolder.***)

Modelling : some basic principles

- **Inheritance**
  - A generalization is a taxonomic relationship between a more general classifier and a more specific classifier.
  - Each instance of the specific classifier is also an indirect instance of the general classifier.
  - **Thus, the specific classifier inherits the features of the more general classifier.**

  (As stated by the OMG UML specification (UML Superstructure Specification, v2.1.1, p. 73).)

Modelling : some basic principles

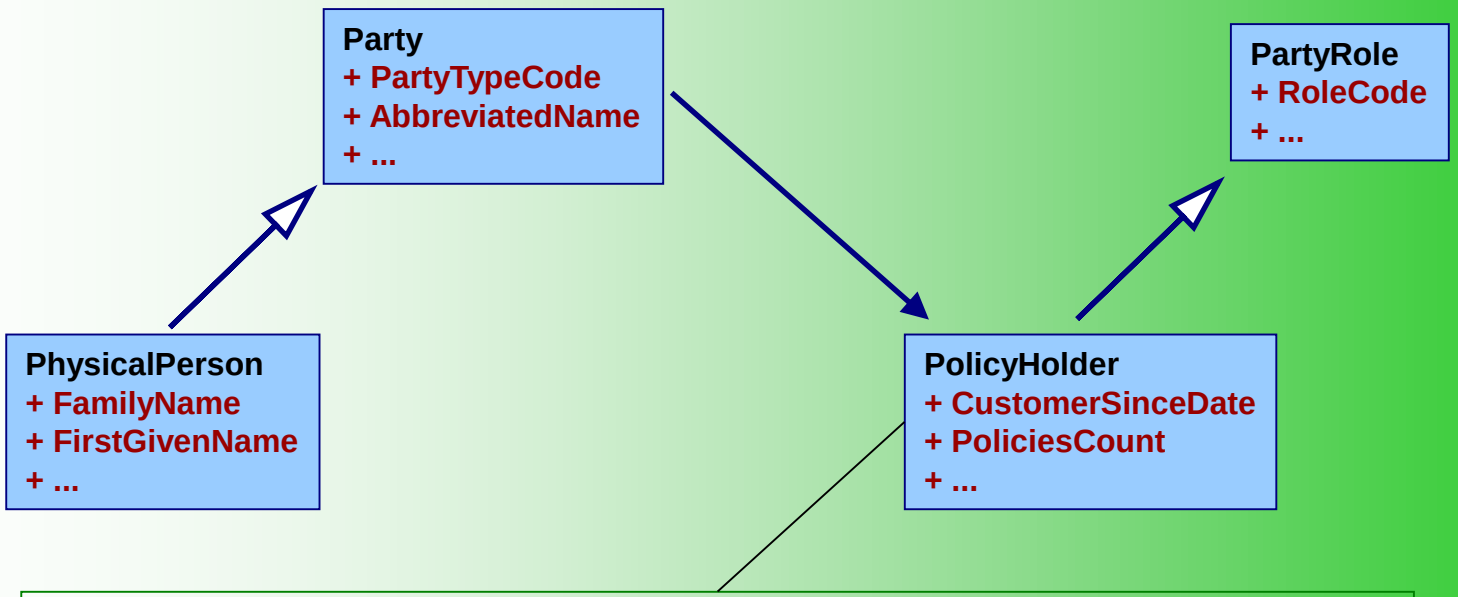


- **Party** is the **Generalization** of PhysicalPerson
- **PhysicalPerson** is the **Extension** of Party
- **PhysicalPerson inherits**
  - the Attributes of Party
  - the Associations of Party

(A PhysicalPerson is a Party, but not every Party is a PhysicalPerson.
Party does not inherit from PhysicalPerson.)

(***A PhysicalPerson is a Party, and so can play the role of PolicyHolder.***)

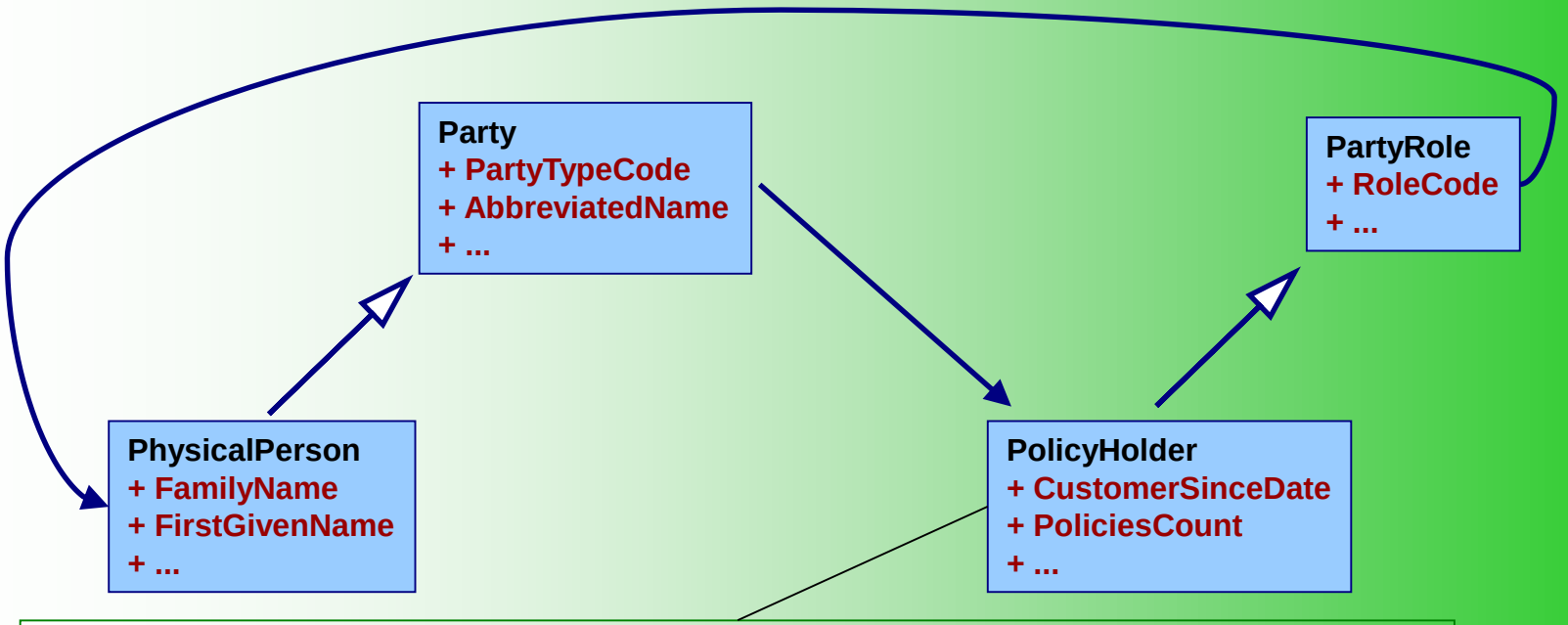
Modelling : some basic principles



- **PartyRole** is the **Generalization** of PolicyHolder
- **PolicyHolder** is the **Extension** of PartyRole
- **PolicyHolder inherits**
  - the Attributes of PartyRole
  - the Associations of PartyRole

(A PolicyHolder role is a PartyRole, but not every PartyRole is a PolicyHolder role. PartyRole does not inherit from PolicyHolder role.)

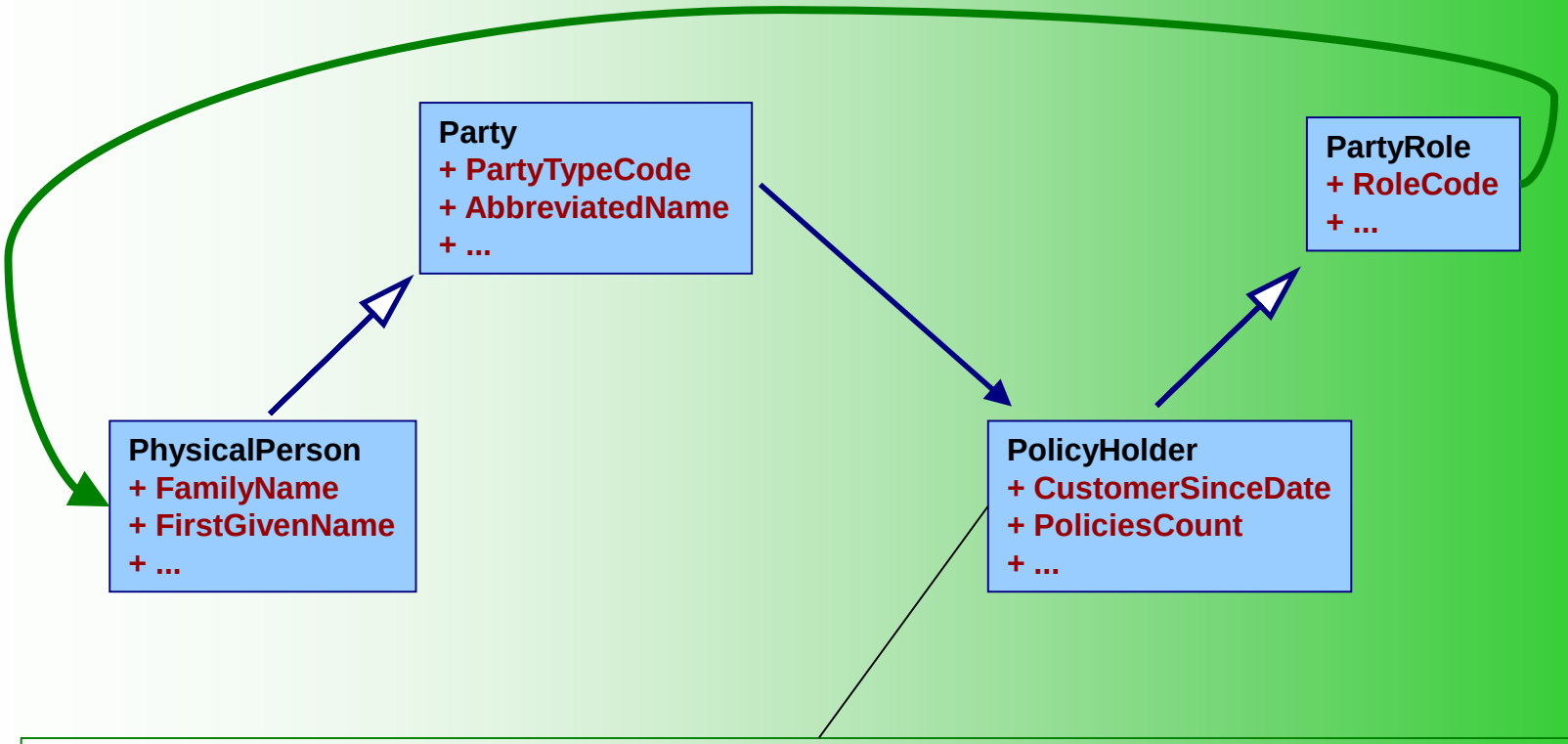
Modelling : some basic principles



- **PartyRole** is the **Generalization** of PolicyHolder
- **PolicyHolder** is the **Extension** of PartyRole
- **PolicyHolder inherits**
  - the Attributes of PartyRole
  - the Associations of PartyRole

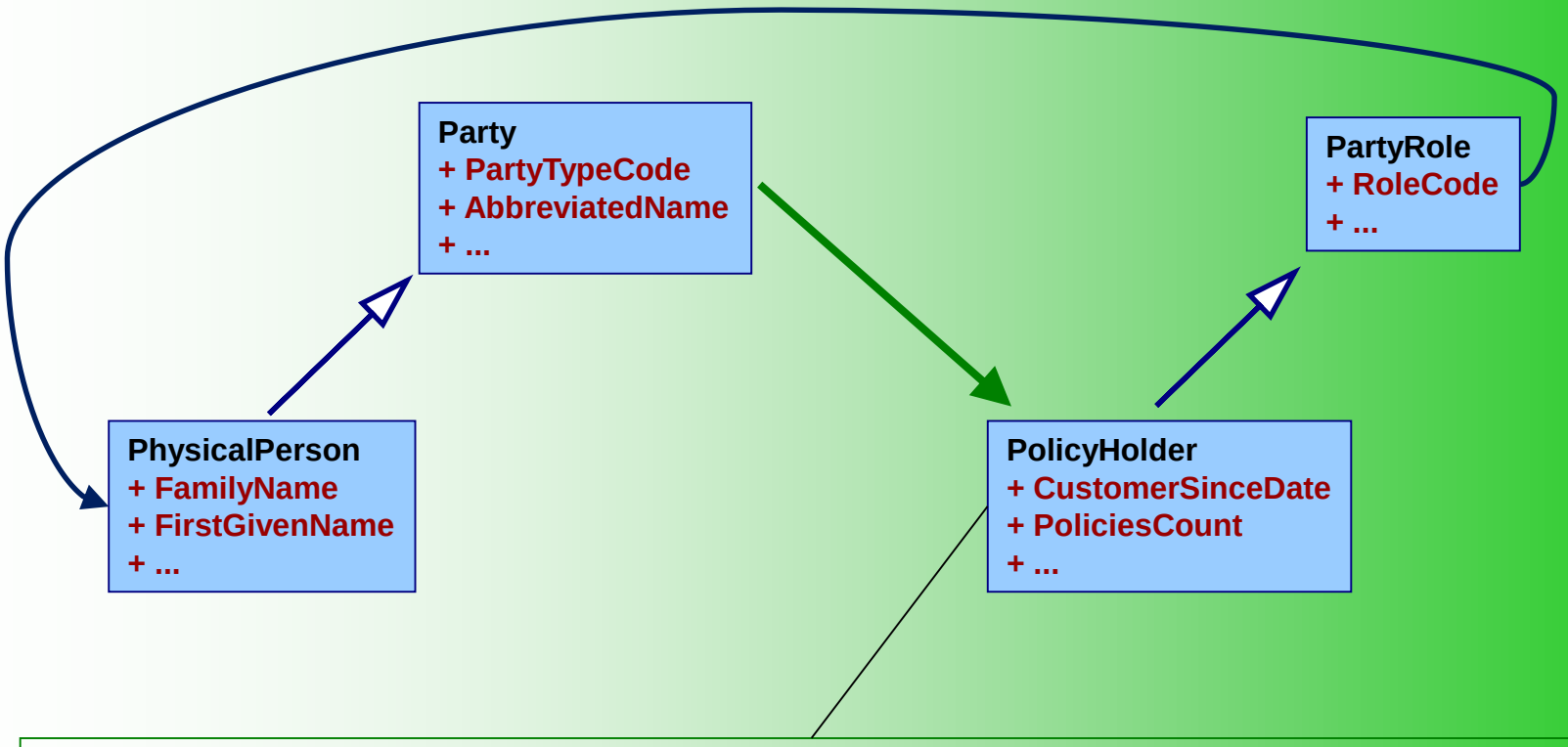
(A PolicyHolder role is a PartyRole, but not every PartyRole is a PolicyHolder role. PartyRole does not inherit from PolicyHolder role.)

Modelling : some basic principles



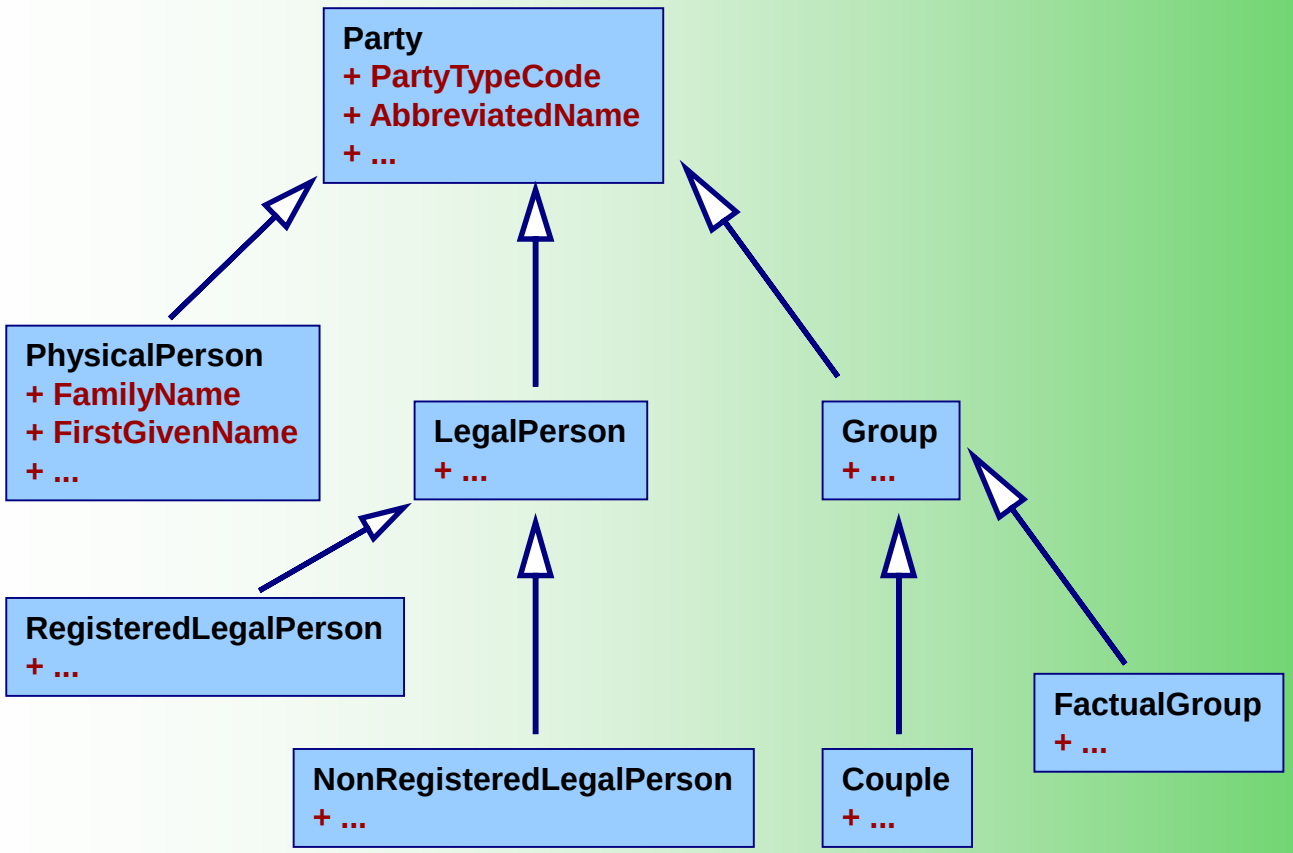
A PolicyHolder role is a PartyRole, which can be played by a PhysicalPerson Party.

Modelling : some basic principles



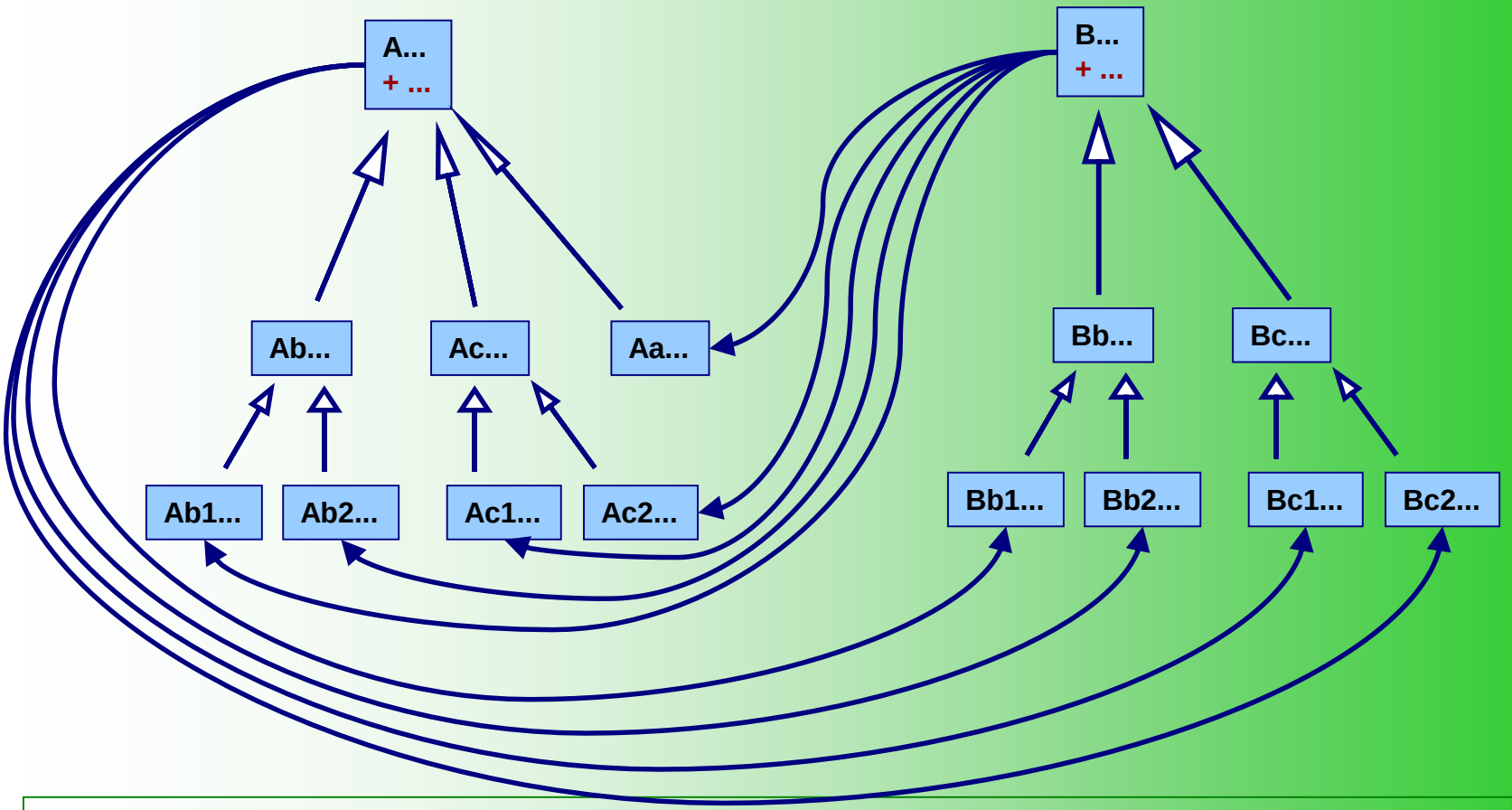
A Physical Person is a Party, which can play a Policyholder Partyrole.

Modelling : some basic principles



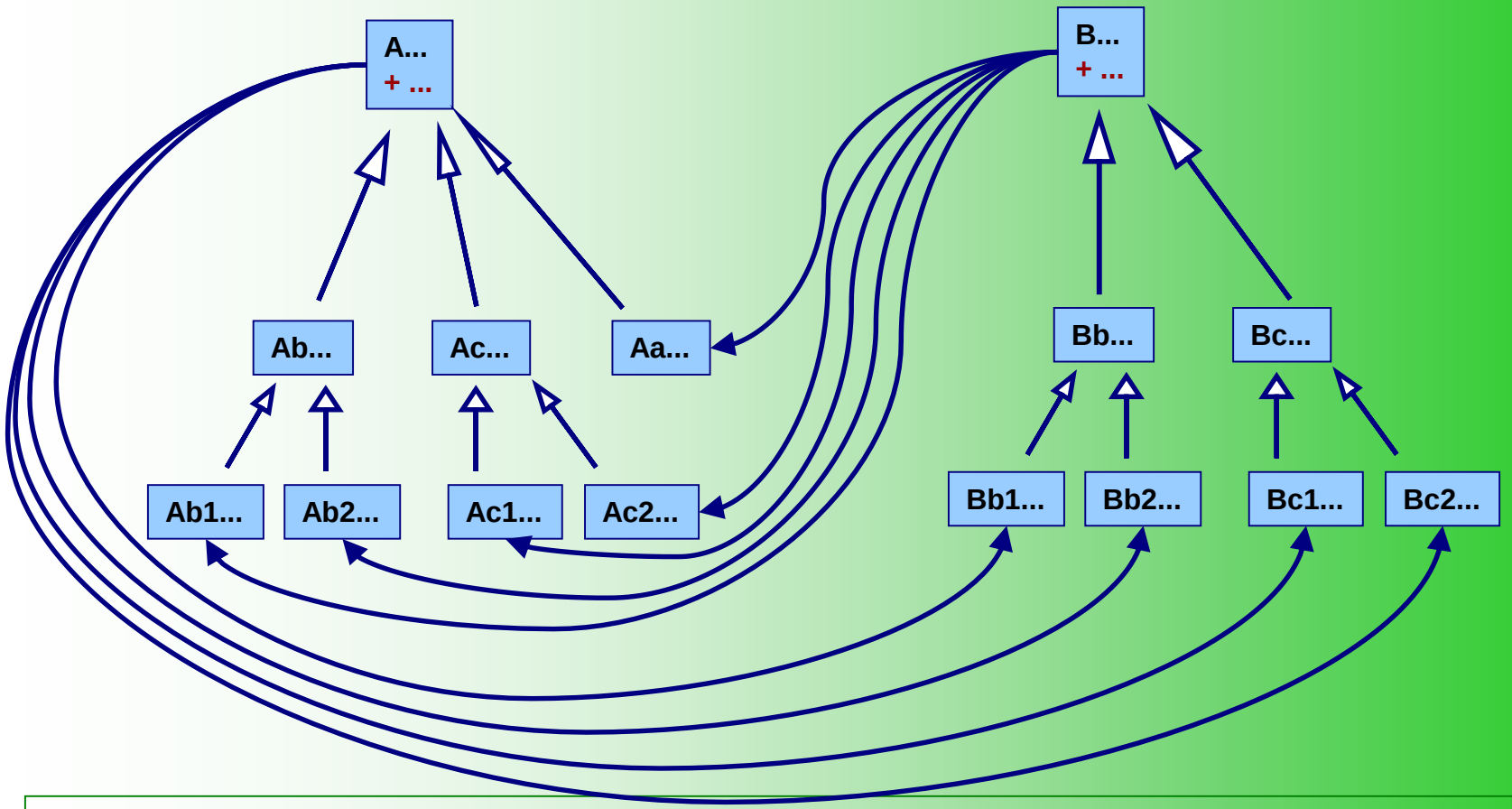
Generalizations/Extensions can become quite complex.

Modelling : some basic principles



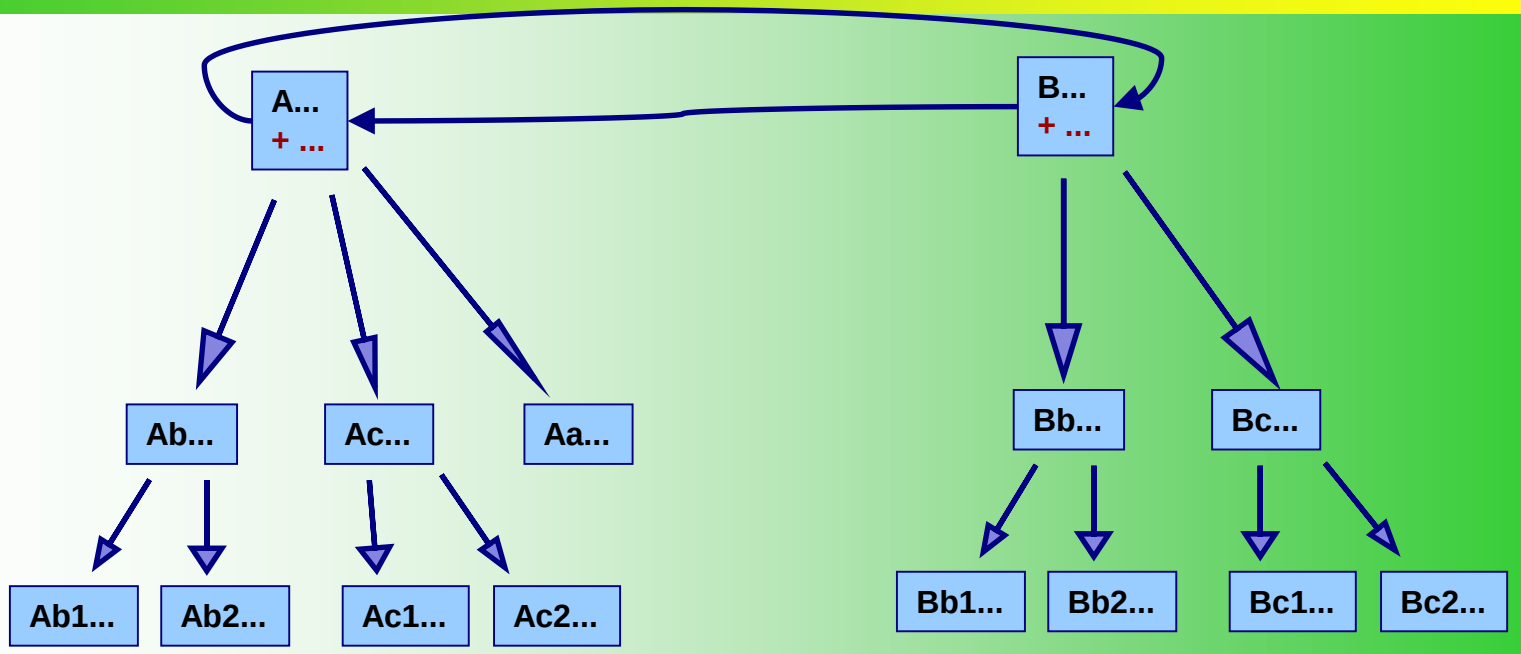
If one creates the Associations from the Generalizations (« high ») to the Extensions (« low »), then the Inheritance is maximized.

Modelling : theory versus practice



You will still end up with A LOT of arrows, and such complexity could easily become too much; it might be better to do it in some other way.

Modelling : theory versus practice

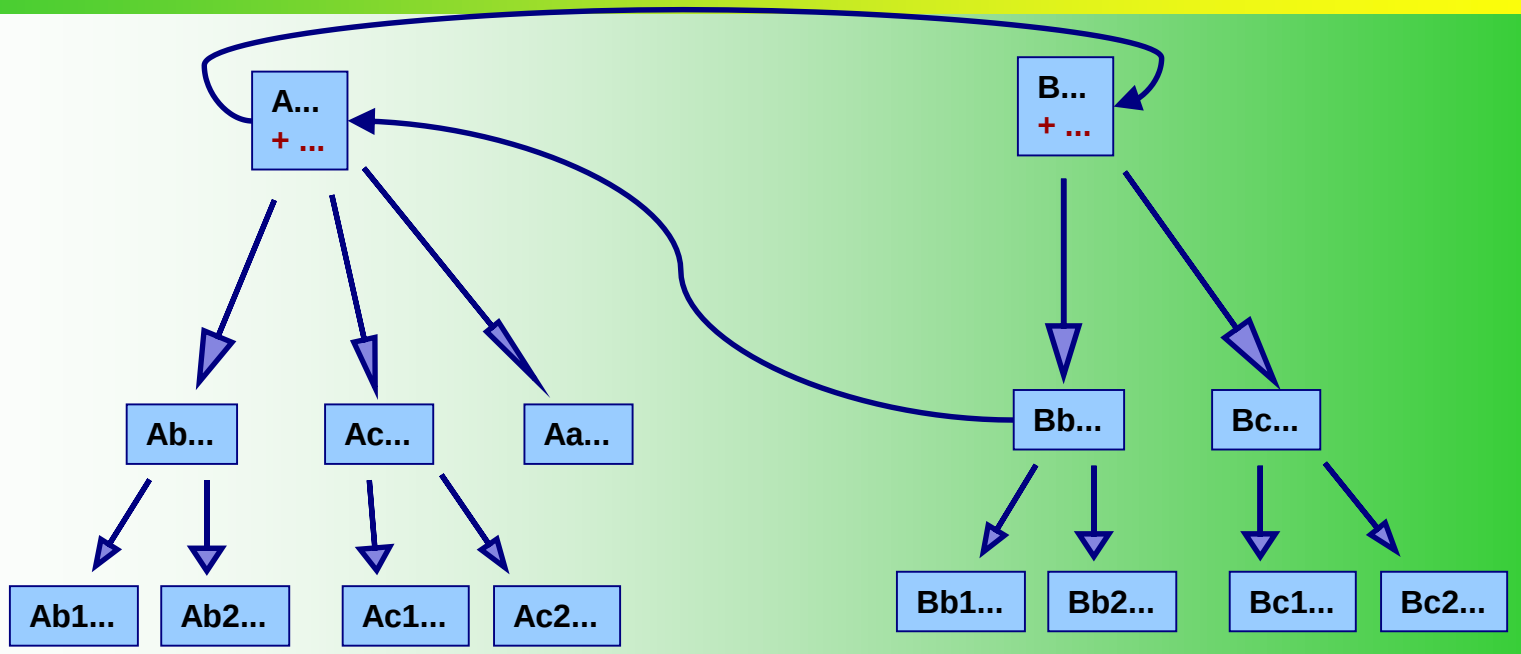


Alternative:

If one creates the Associations on the level of the Generalizations (« high »)
- then the number of Associations is minimised;
- then Inheritance doesn't work.

Replacing Inheritance with « top-down added choices out of sets of clusters of information »
- could work;
- but is not very intuitive, nor accepted by user communities.

Modelling : theory versus practice



Alternative, <u>consequence:</u>

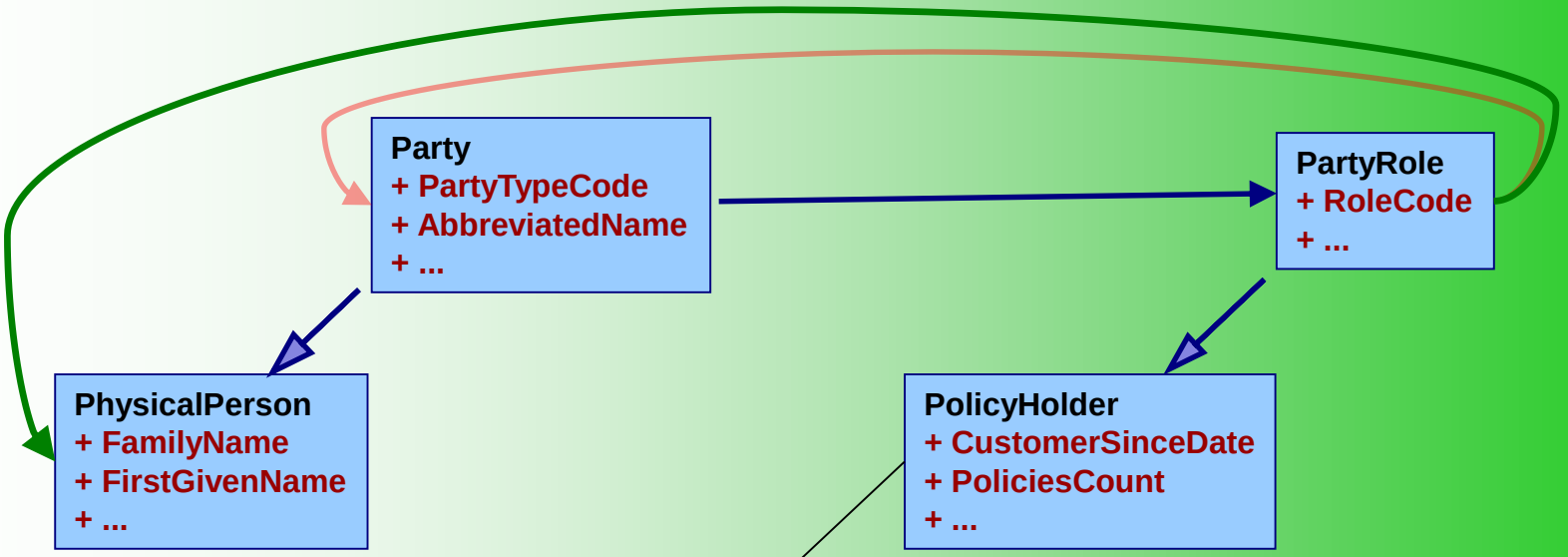
Wherever Associations do start from, it is important that they end/enter on the level of the Generalization, if not the « higher up » attributes cannot be reached...

Replacing Inheritance with « top-down added choices out of sets of clusters of information »

could still work;

but again, is not very intuitive, nor accepted by user communities

Modelling : in practice, this does the job

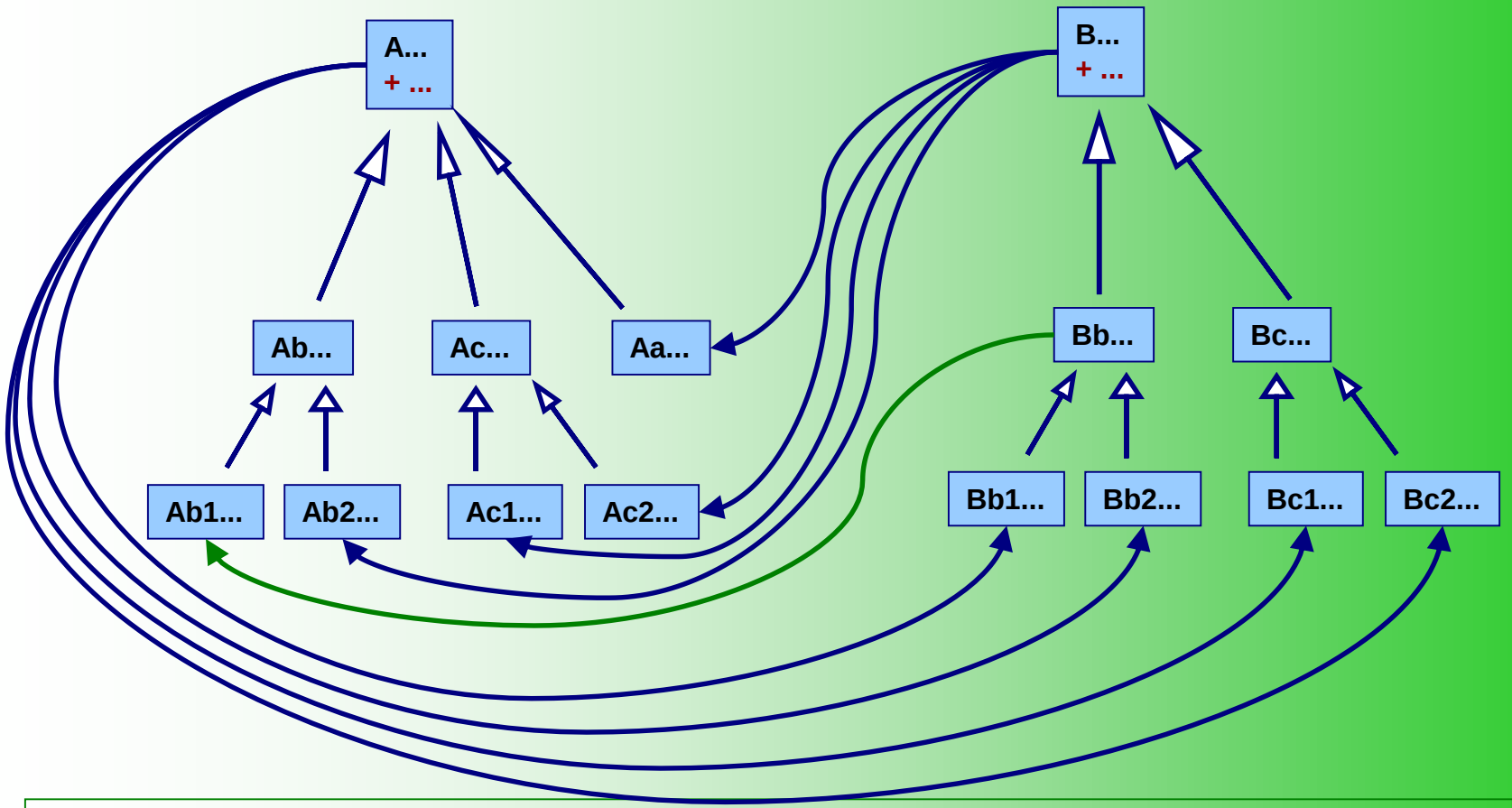


- **PartyRole** is the **Generalization** of PolicyHolder
- **PolicyHolder** is the **Extension** of PartyRole
- **PolicyHolder inherits**
  - the Attributes of PartyRole
  - the Associations of PartyRole

  (A PolicyHolder role is a PartyRole, but not every PartyRole is a PolicyHolder role. PartyRole does not inherit from PolicyHolder role.)

A PolicyHolder role is a PartyRole, which can be played, not by some abstract Party, but by a PhysicalPerson – amongst other.

Modelling : theory versus practice



Starting some Association out of some Extension and not out of the top-level Generalization,
is like positively **filtering** potential associations;
is like negatively **filtering** sibling extensions.
This will certainly not reduce the number of associations; to the contrary – there might be more...