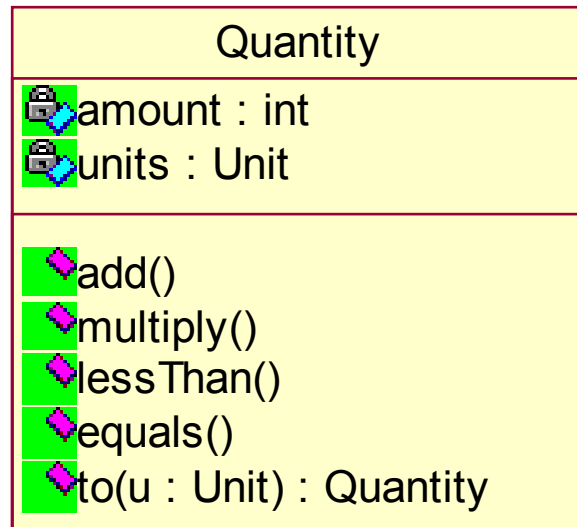
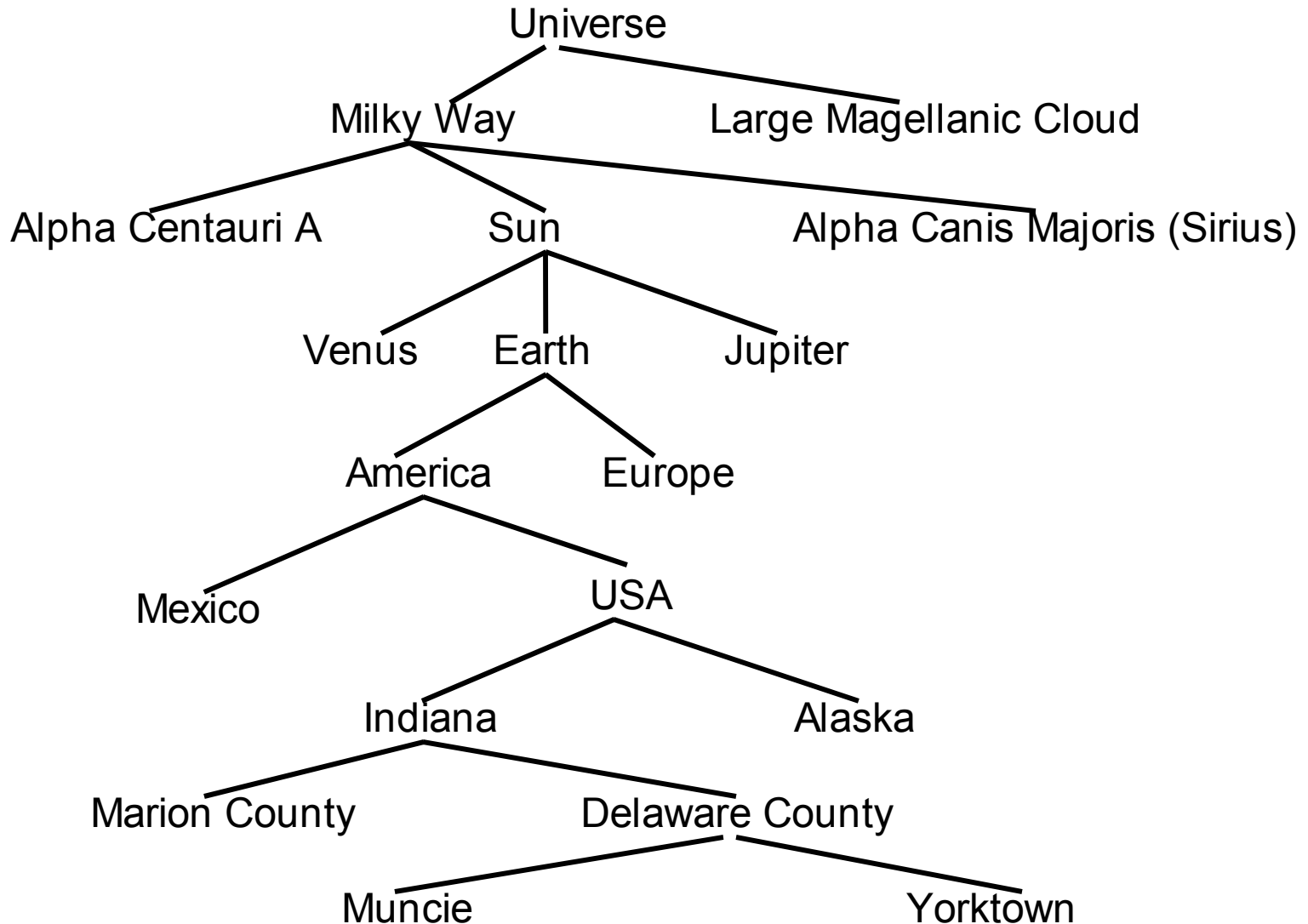


Analysis Pattern: Quantity



- For example: money, weight, length, voltage...
- Allows for unit conversion
 - feet to meters
 - euros to dollars
- Note: never represent amounts of money as floating point numbers (float, double in Java)

Analysis Pattern: Hierarchy



Analysis Pattern: Hierarchy

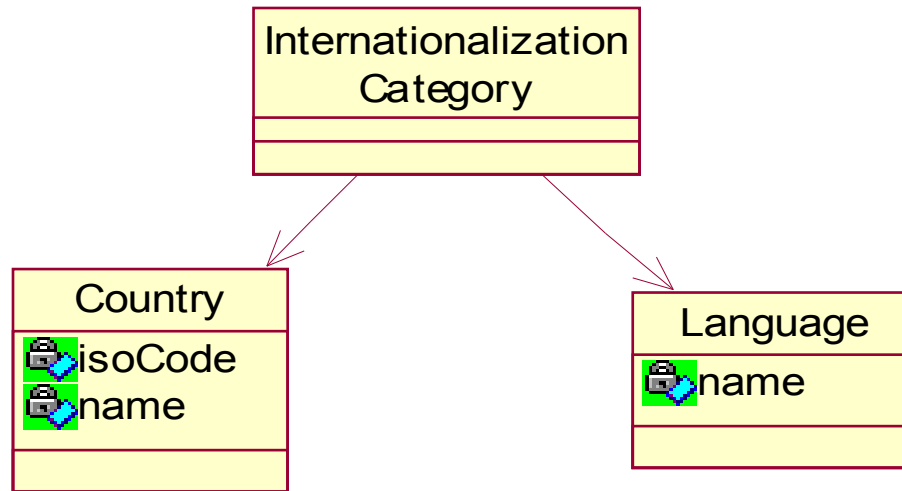
- Paths in some hierarchies:
 1. Universe – Earth – USA – Indiana – Delaware County – Muncie
 2. Organism - Animal - Chordate – Mammal – Primate – Human
 3. Director of Undergrad Programs – Chair of CS – Dean of Science and Humanities – President of BSU
- Hierarchy relation:
 - 1 – is-part-of,
 - 2 – is-a-kind-of,
 - 3 – reports-to
- A hierarchy is
 - a tree with a specific, common relation along its edges
 - thus: except for the root, everything has exactly one parent

Hierarchy

- Year – Month – Week – Day
 - Problematic!
- Year – Month – Day
- Week – Day
 - These are really two hierarchies!

- USA – Indiana – Delaware County - Muncie
 - what about unincorporated areas of Delaware County?
 - what about cities that are in several counties (like New York)?
 - what about the District of Columbia?

Analysis Pattern: Internationalization

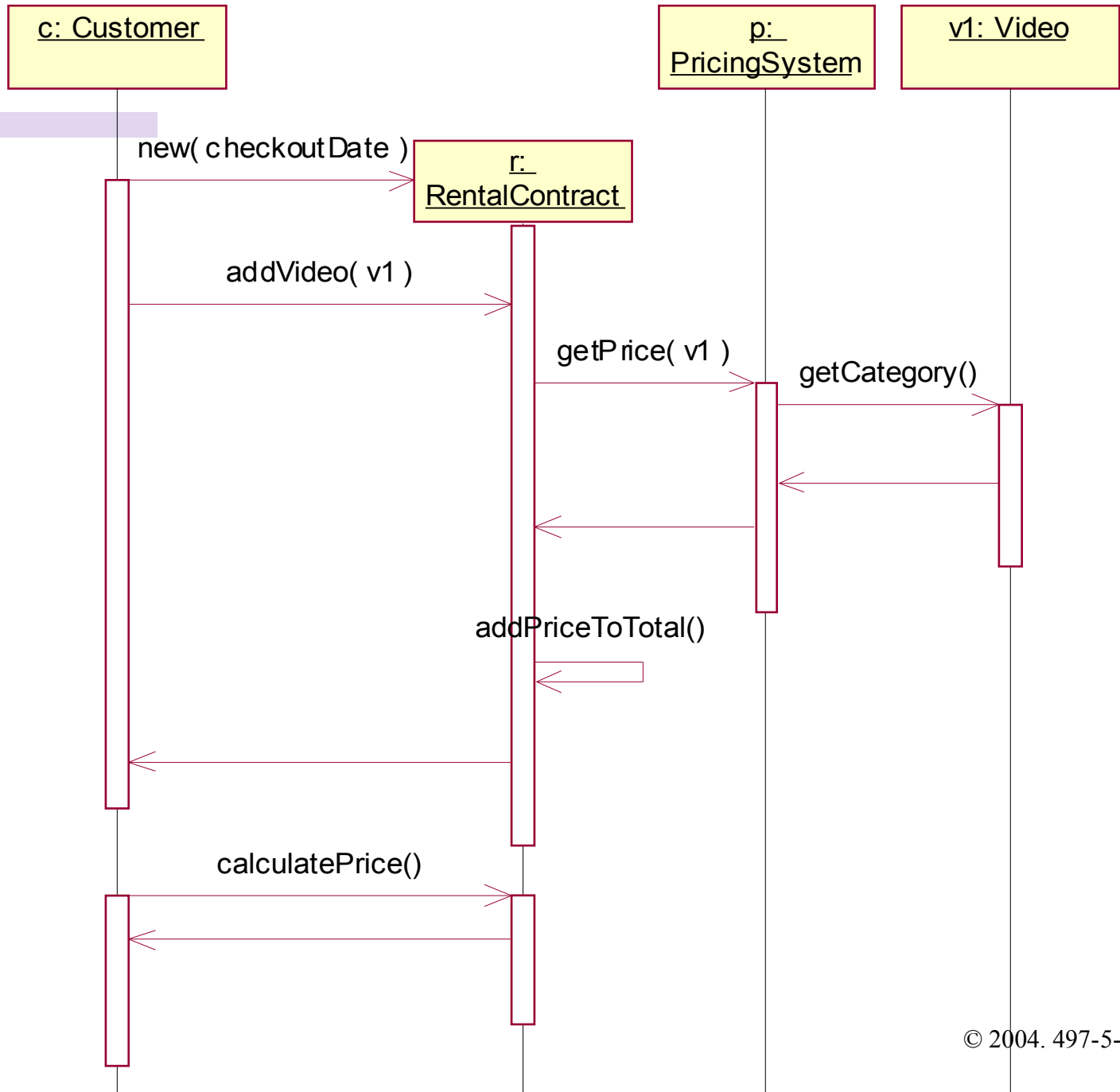


- Problem: software used in many countries needs to support:
 - languages
 - spelling conventions
 - currency symbols
- Examples:
 - USA / English
 - USA / Spanish
 - UK / English

4.3 Interaction Diagrams

- Extended UML object diagrams with messages
- Show behavior of a set of objects
 - dynamic
 - have a time scale
- Like all object diagrams: can't be complete
 - show only an example of the behavior
- Two kinds of UML interaction diagrams
 - sequence diagrams
 - communication diagrams

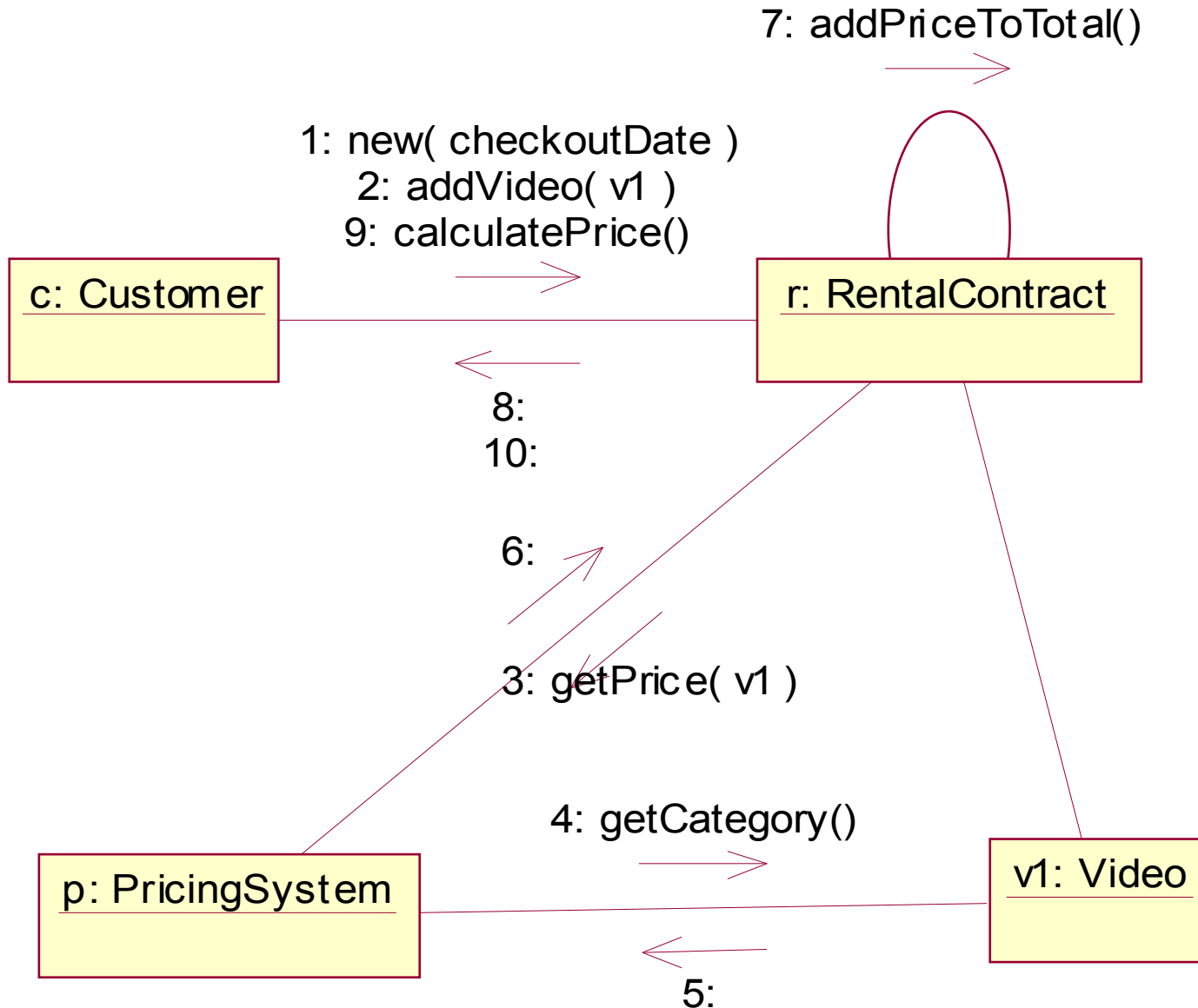
Sequence Diagram



Elements of Sequence Diagrams

- Participants
 - typically objects
 - life lines: time goes from top to bottom
 - there may be several instances of one class
 - boxes on life lines show if object is active
- Messages
 - analogous to method calls in a program
 - can have parameters
 - special messages
 - new — shown by position of object
 - delete — shown through big X
 - return messages
 - self-calls
- Additional syntax for asynchronous calls (concurrent)
 - concurrent programming not covered in this class

Communication Diagram



Elements of Communication Diagrams

- Formerly (UML 1): “Collaboration Diagrams”
- Participants
 - like objects
- Links
- Messages
 - sorted by direction on each link
 - numbered to show sequence
 - better -- numbered with dots: 1, 1.1, 1.2, ...
 - shows nesting of method calls
- Communication diagrams are similar to object diagrams

Sequence and Communication Diagrams

- Are exactly equivalent
 - can be automatically transformed into each other
- Sequence diagrams well-suited for showing time sequence of messages
- Communication diagrams well-suited for showing relations among objects
- Use in analysis
 - to show typical interactions in a use case
- Use in design
 - to show complex interactions between a set of objects
- Limitations
 - no time scale
 - no branching, looping
 - can only show sample behavior