UML

Feliz Gouveia
UFP

Introducion

- Unified Modeling Language
- International standard for software analysis and design
- Currently a huge standard, covering most of the documentation needs
- 14 UML diagram types

UML diagrams

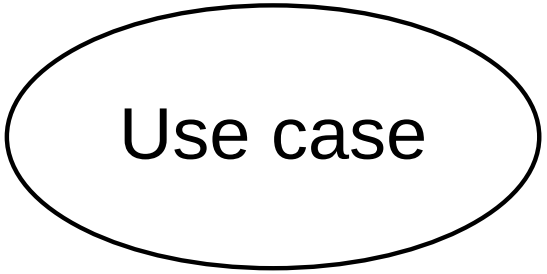
- Class diagram
- Component diagram
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram
- Composite structure diagram
- Use case diagram
- Activity diagram
- State machine diagram
- Sequence diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

Use case diagrams

- Are used during requirements elicitation and analysis as a graphical means of representing the functional requirements of the system.
- Use cases are very helpful for writing acceptance test cases.
- Consist of 4 objects:
  - Use cases
  - Actors
  - System
  - Package

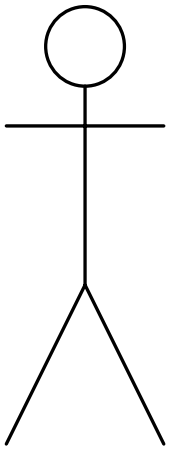
Use case diagrams: use case

- A use case typically represents a major piece of functionality that is complete from beginning to end.
- Represents a function or an action within the system.



Use case diagrams: actors

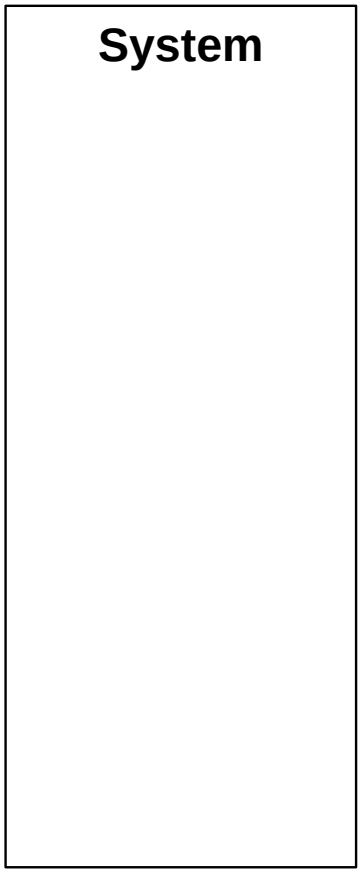
- An actor represents whoever or whatever (person, machine, or other) interacts with the system.
- The actor is not part of the system itself and represents anyone or anything that must interact with the system to:
  - Input information to the system;
  - Receive information from the system; or
  - Both input information to and receive information from the system.



Actor

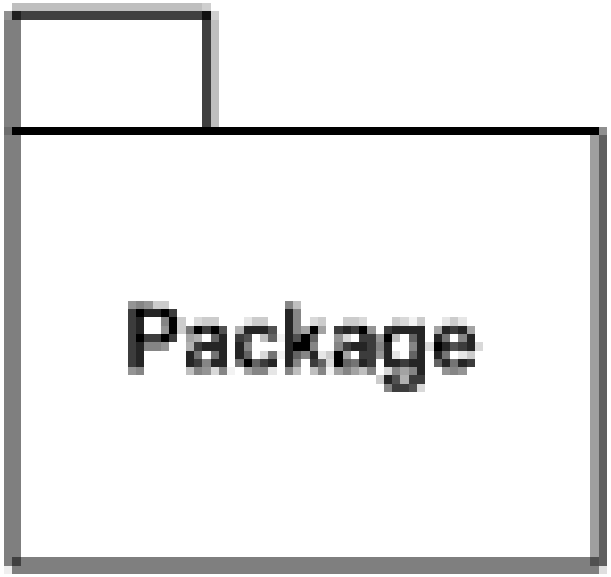
Use case diagrams: system

- System is used to **define the scope of the use case** and drawn as a rectangle.



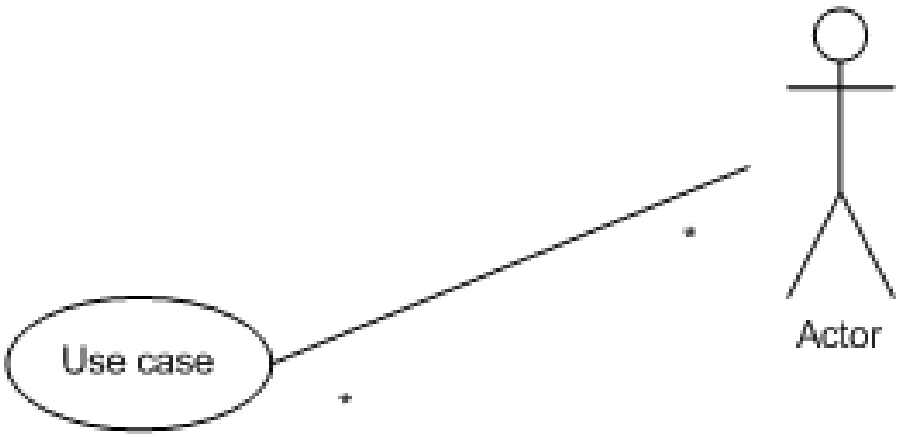
Use case diagrams: package

- Package is another optional element that is extremely useful in complex diagrams. Similar to class diagrams, packages are **used to group together use cases**.



Use cases: notation

- Actor comunicates with the system



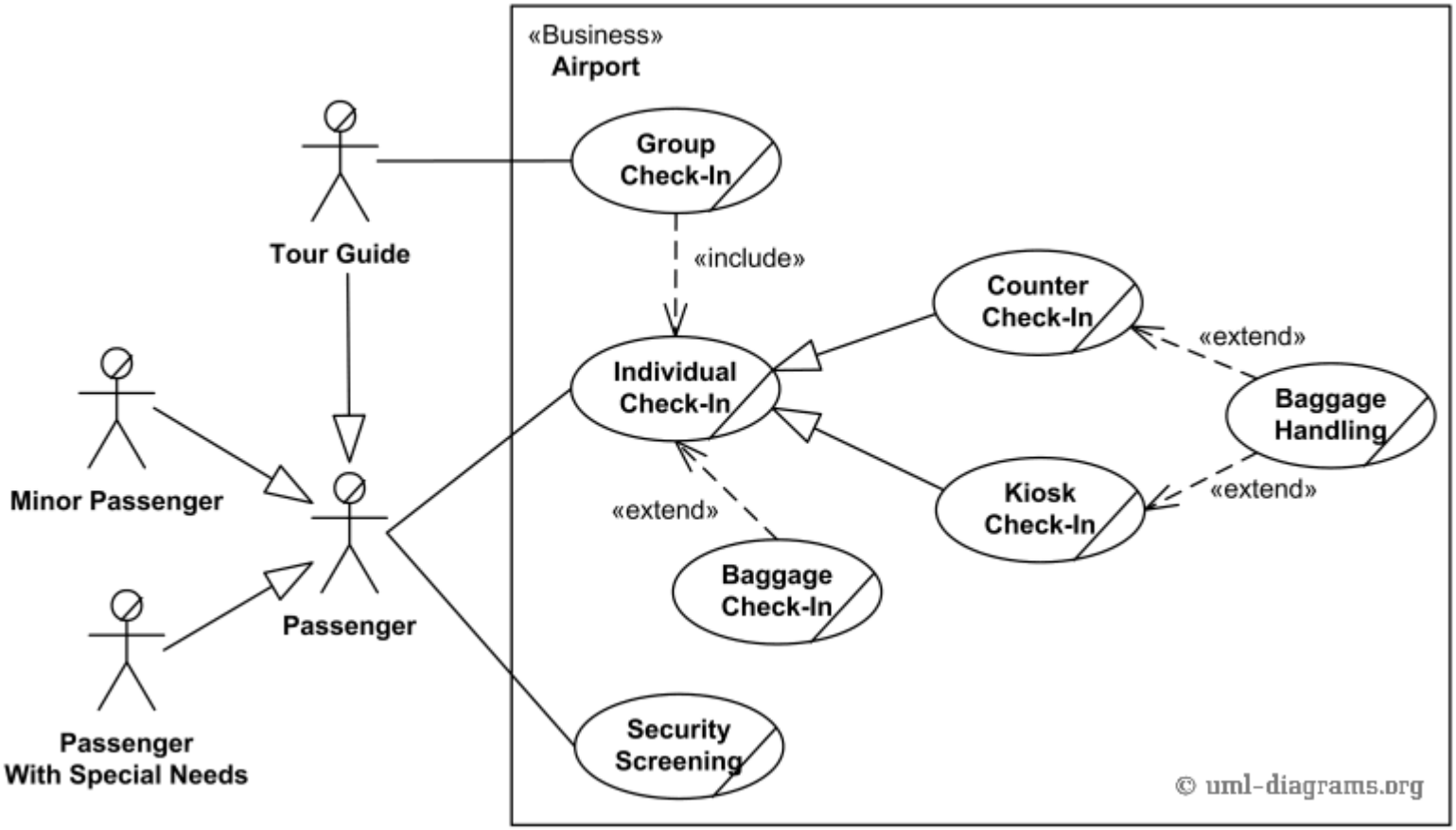
Use case diagrams: relationships

- There are five types of relationships in a use case diagram. They are:
  - Association between an actor and a use case
  - Generalization of an actor
  - Extend relationship between two use cases
  - Include relationship between two use cases
  - Generalization of a use case

Use case: relationships

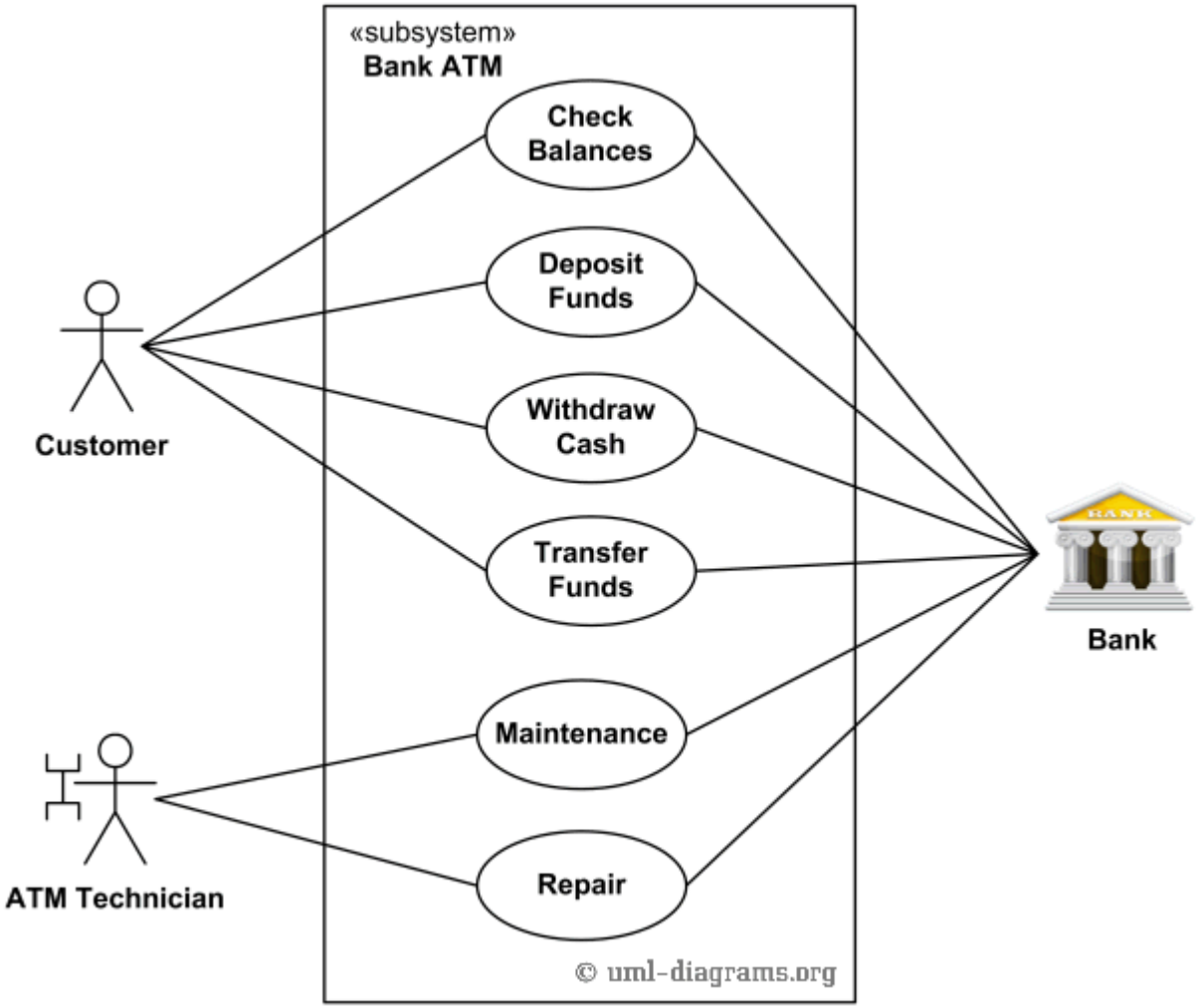
- A use case can “include” another use case; this is useful to avoid repeating the description of a behavior; e.g. if several use cases require a login, it’s best to include the login use case;
- A use case can “extend” another use case; this can be used for variations on a use case, that only happen under certain circunstances;
- Note that an include use case is always completed, but an exclude use case is sometimes completed

Use cases: example

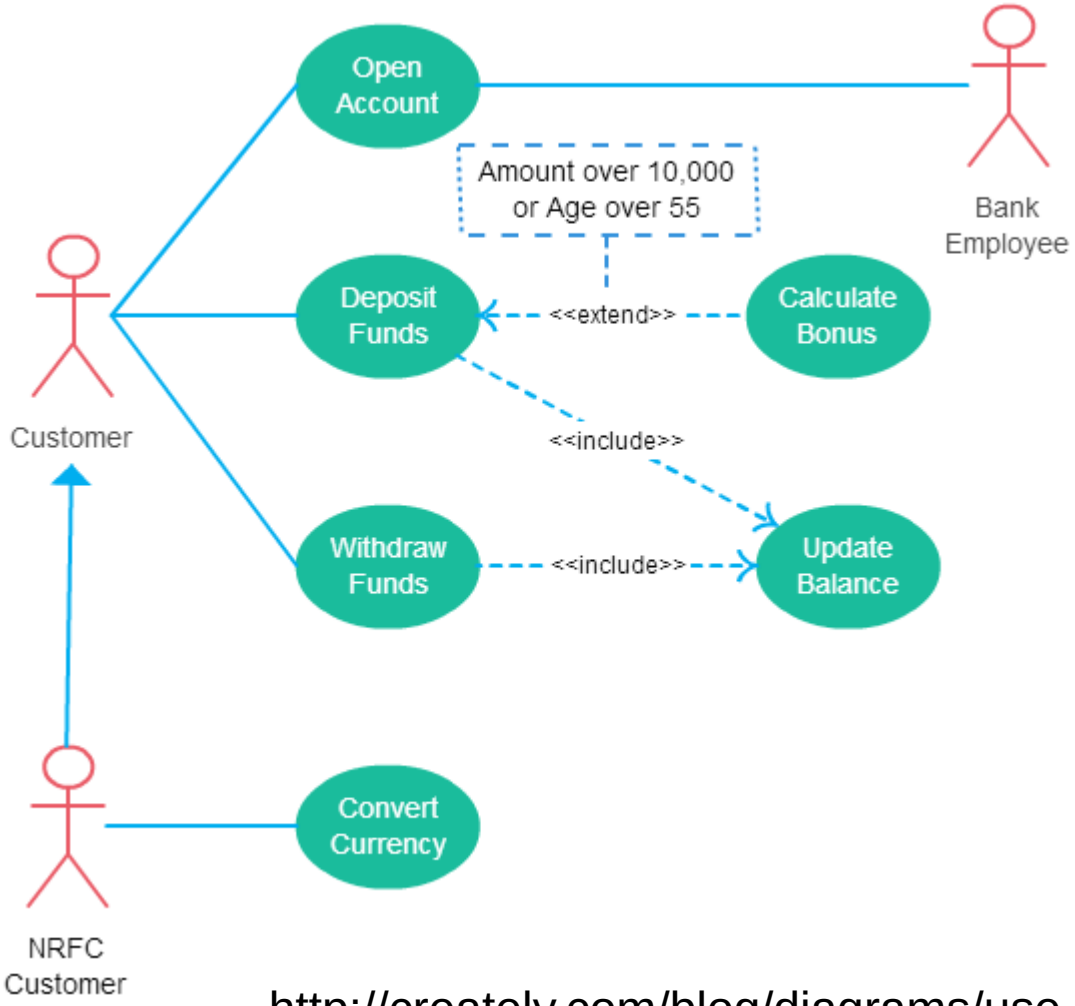


http://www.uml-diagrams.org/

Use cases: example



Use cases: example



http://creately.com/blog/diagrams/use-case-diagram-relationships/

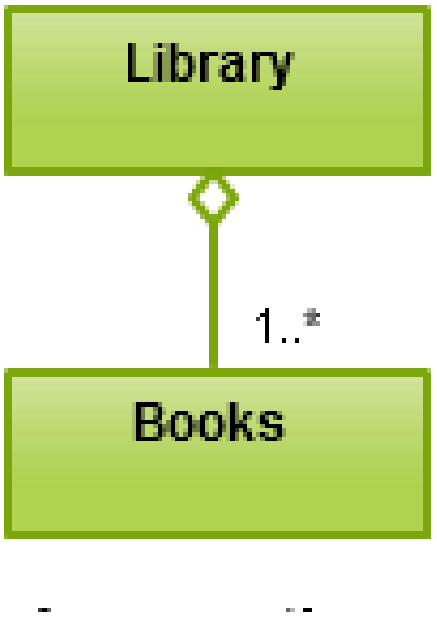
Class diagram

- Class diagrams are used in both the analysis and the design phases.
- They provide detailed information about the structure and the behavior of the classes
- There are two main types of relationships between classes:
  - Inheritance
  - Association

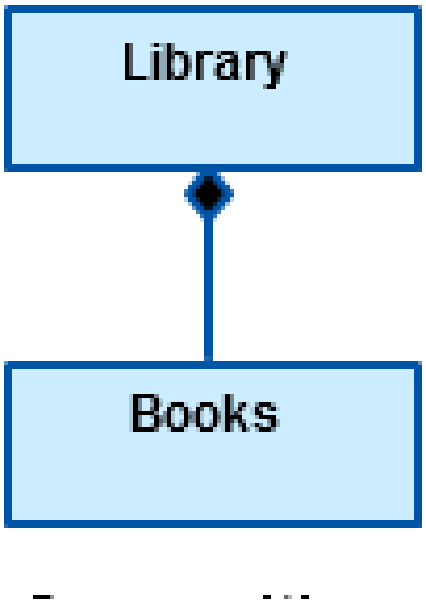
Class relationships

- Inheritance: one class is more general than others
- Association:
  - Aggregation: one object is part of another object but their lifetimes are independent (a department has employees)
  - Composition: one object is composed of other objects (an invoice is composed of invoice lines) and those can only exist as long as the container object exists
  - Association has multiplicity

Class relationships



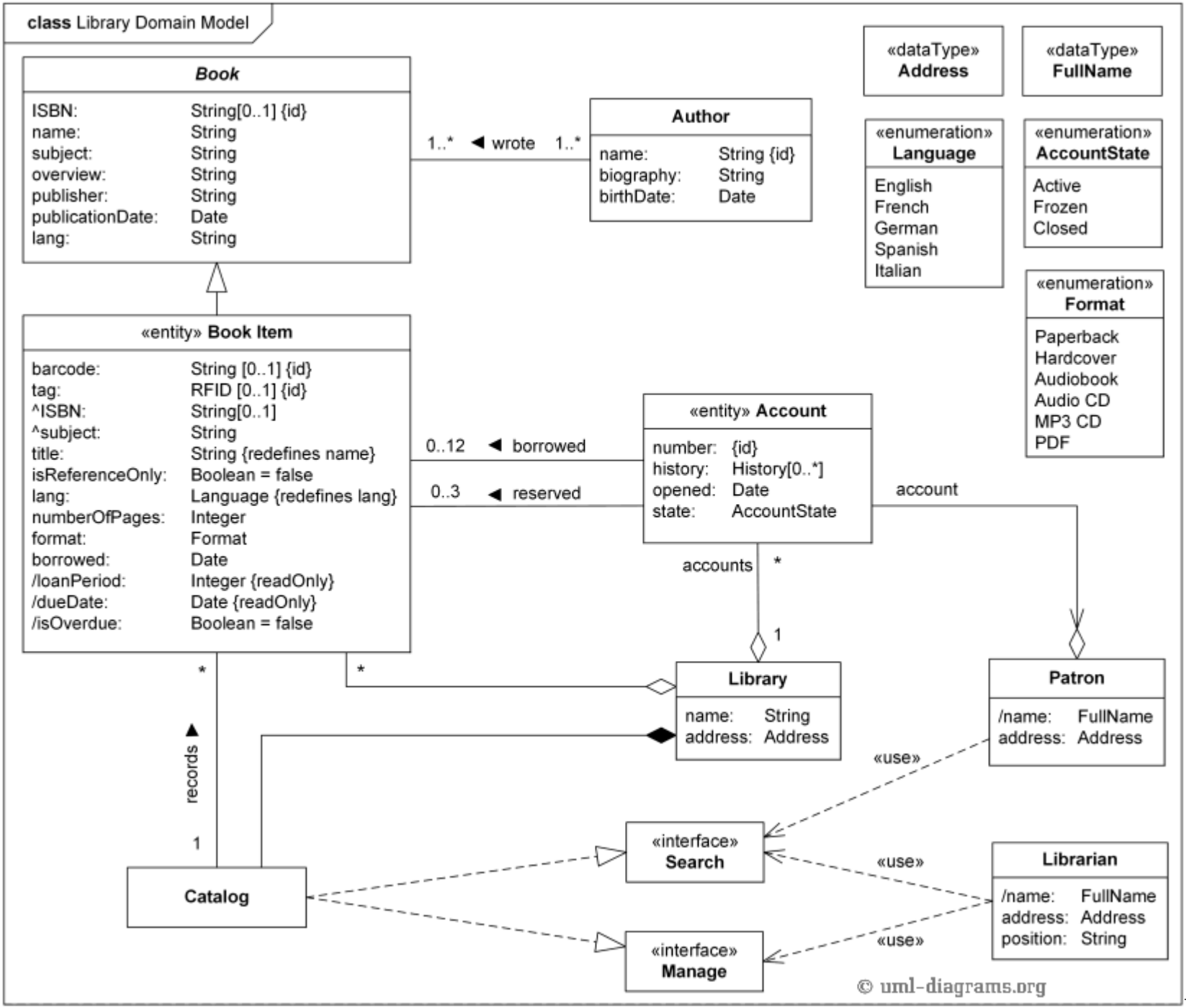
aggregation



composition

Class relationships: multiplicity

- One-to-one and mandatory: 1
- One-to-one and optional: 0..1
- One-to-many and mandatory: 1..*
- One-to-many and optional: *
- With lower bound l and upper bound u: l..u
- With lower bound l and no upper bound: l..*


class Library Domain Model
Book
ISBN: String[0..1] {id}
name: String
subject: String
overview: String
publisher: String
publicationDate: Date
lang: String
1..* wrote 1..*
Author
name: String {id}
biography: String
birthDate: Date
«dataType» Address
«dataType» FullName
«enumeration» Language
English
French
German
Spanish
Italian
«enumeration» AccountState
Active
Frozen
Closed
«enumeration» Format
Paperback
Hardcover
Audiobook
Audio CD
MP3 CD
PDF
«entity» Book Item
barcode: String [0..1] {id}
tag: RFID [0..1] {id}
^ISBN: String[0..1]
^subject: String
title: String {redefines name}
isReferenceOnly: Boolean = false
lang: Language {redefines lang}
numberOfPages: Integer
format: Format
borrowed: Date
/loanPeriod: Integer {readOnly}
/dueDate: Date {readOnly}
/isOverdue: Boolean = false
0..12 borrowed
0..3 reserved
«entity» Account
number: {id}
history: History[0..*]
opened: Date
state: AccountState
account
accounts
*
1
*
*
records
1
Library
name: String
address: Address
Patron
/name: FullName
address: Address
«use»
«interface» Search
«use»
Librarian
/name: FullName
address: Address
position: String
Catalog
«interface» Manage
«use»
© uml-diagrams.org