Example – Class analysis with Noun

Noun Identification: a Library Example

The library contains books and journals. It may have several copies of a given book. Some of the books are reserved for short-term loans only. All others may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

The system must keep track of when books and journals are borrowed and returned, and enforce the rules.

Ref: CS5150-Cornel University 16-prog-design.pdf

Example – Class analysis with Noun

Noun Identification: a Library Example

The library contains books and journals. It may have several copies of a given book. Some of the books are reserved for short-term loans only. All others may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

The system must keep track of when books and journals are borrowed and returned, and enforce the rules.

Example – Class analysis with Noun

Candidate Classes

Noun	Comments	Candidate
Library	*the name of the system*	no
Book		yes
Journal		yes
Copy		yes
ShortTermLoan	*event*	no (?)
LibraryMember		yes
Week	*measure*	no
MemberOfLibrary	*repeat of LibraryMember*	no
Item	*book or journal*	yes (?)
Time	*abstract term*	no
MemberOfStaff		yes
System	*general term*	no
Rule	*general term*	no

Example – Class analysis with Noun

Relations between Classes

Book	is an	Item
Journal	is an	Item
Copy	is a copy of a	Book
LibraryMember		
Item		
MemberOfStaff	is a	LibraryMember

Is Item needed?

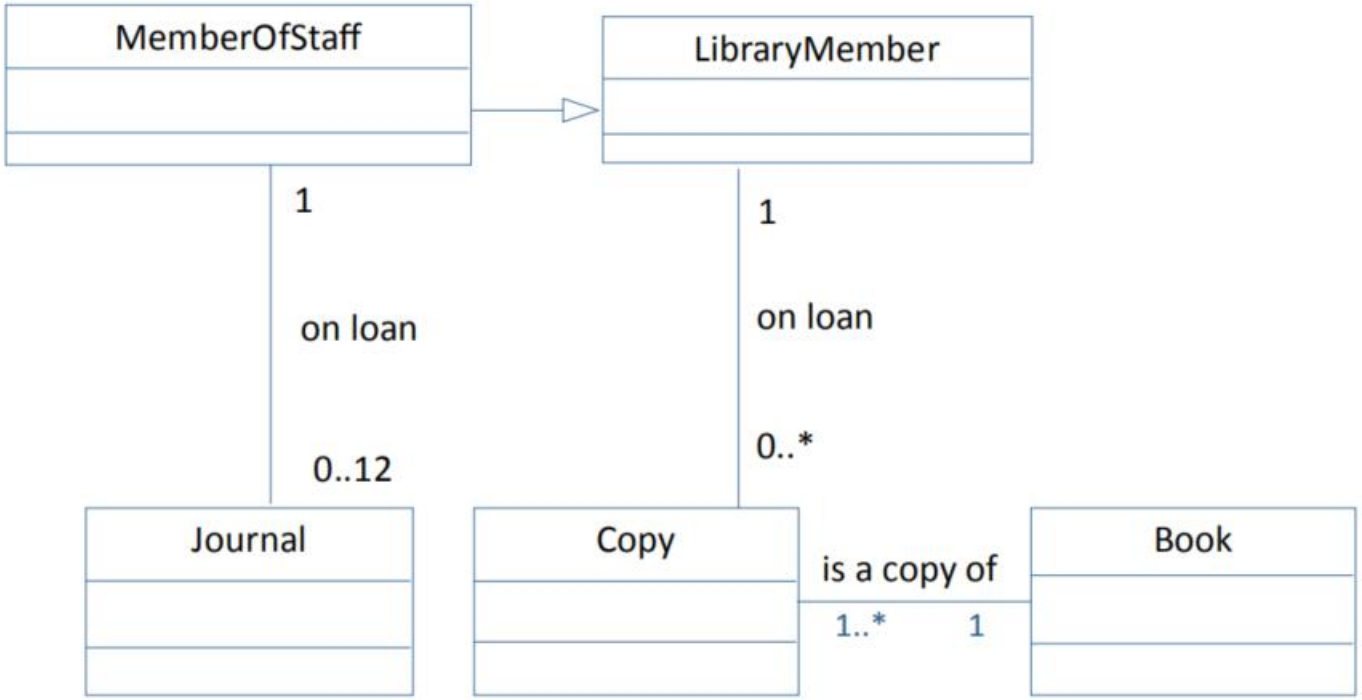
Methods

LibraryMember	borrows	Copy
LibraryMember	returns	Copy
MemberOfStaff	borrows	Journal
MemberOfStaff	returns	Journal

Item not needed yet.

Example – Class analysis with Noun

A Possible Class Diagram



Example – Class analysis with Noun

From Candidate Classes to Completed Design

Methods used to move to final design

Reuse: Wherever possible use existing components, or class libraries. They may need extensions.

Restructuring: Change the design to improve understandability, maintainability, etc. Techniques include merging similar classes, splitting complex classes, etc.

Optimization: Ensure that the system meets anticipated performance requirements, e.g., by changed algorithms or restructuring.

Completion: Fill all gaps, specify interfaces, etc.

Design is iterative

As the process moves from preliminary design to specification, implementation, and testing it is common to find weaknesses in the program design. Be prepared to make major modifications.

THE END