Cheatography

Domain Modelling Cheat Sheet

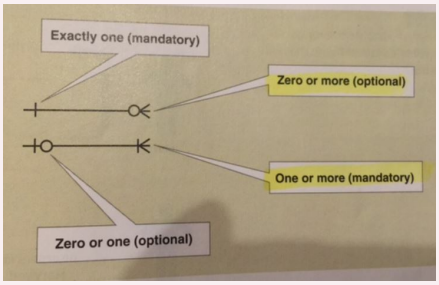
by Natalie Moore (NatalieMoore) via cheatography.com/19119/cs/3748/

"Things" in the Problem Domain

Problem domain = The specific area (or domain) of the users' business. Scope of the system.
Examples: products, sales, shippers, customers, invoices, payments
"Things" eventually modelled.

Called "classes" (UML) or "entities"

Cardinality symbols ERD relationships



The brainstorming technique

1. Identify a user and a set of use cases.
2. Brainstorm with user to ID things involved
3. Use the types of things (categories) to systematically ask questions about potential things, such as the following: Are there any tangible things you store information about? Are there any locations involved? Are there roles played by people that you need to remember?
4. Continue to work with all types of users and stakeholders to expand the brainstorming list.
5. Merge the results, eliminate any duplicates, and compile an initial list.

Joint effort between the analyst and the users.

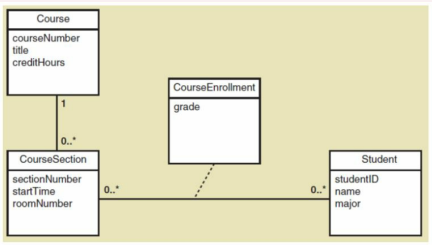
Determine

Determine entities (things in the domain)
Attributes (of entities)
Keys (Primary, Foreign)
Data types (integer, string, etc)
Relationships (which entities are related)
Cardinality (how are entities related)

Details about Entities/Classes

Attribute: describes one piece of information about each instance of the class
Customer has first name, last name, phone number
Identifier or key: One attribute uniquely identifies an instance of the class. Required for data entities, optional for domain classes. Customer ID identifies a customer
Compound attribute: Two or more attributes combined into one structure to simplify the model.
Object: Class is a type of thing. Object is a specific instance of the class. Each instance has its own values for an attribute
Association (UML) / Relationship (ERD): a naturally occurring relationship between classes (UML term). # rep by Multiplicity/Cardinality. 1 to 1 or 1 to many.

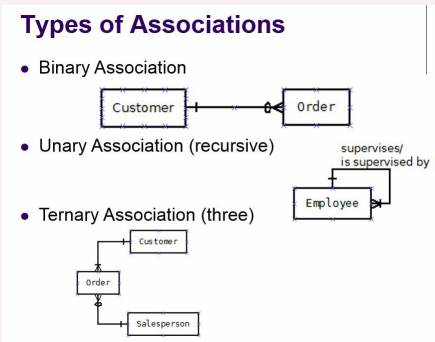
Domain Class Model Diagram with an association



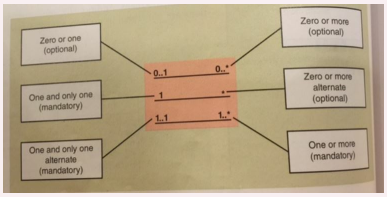
Types of things

Tangible	e.g. book, plane, car, hat, document, worksheet
Role	Employee, Customer, doctor
Org unit	Division, department, team
Devices	Sensor, timer, machine, printer
Sites / locations	Warehouse, branch, store, desk
Incident / event / interaction	Fligh, call, logon, order, payment

Types of associations



UML notation for multiplicity of associations





By **Natalie Moore** (NatalieMoore)

cheatography.com/nataliemoore/
www.clipto.com/transcribe-audio-video-to-text-free?via=natalie

Published 30th May, 2015.
Last updated 12th May, 2016.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

The Noun Technique

1. Using the use cases, actors, and other information about the system — including inputs and outputs — identify all nouns (names).
2. Using other information from existing systems, current procedures, and current reports or forms, add items or categories of information needed.
3. As this list of nouns builds, you will need to refine it. Ask these questions about each noun to help you decide whether you should include it:
- Ask questions, such as is it important and inside the scope, to see if it should be included.
- Ask questions to see if it really should be excluded, such as is it only a report or an input or is it an attribute.
- Ask questions to see if it needs further research. In other words that you cannot answer whether it needs to be included or excluded.
4. Create a master list of all nouns identified and then note whether each
one should be included, excluded, or researched further.
5. Review the list with users, stakeholders, and team members and then refine the list of things in the problem domain.

Mechanical approach to identifying classes



By **Natalie Moore**
(NatalieMoore)

cheatography.com/nataliemoore/
www.clipto.com/transcribe-audio-video-to-text-free?via=natalie

Published 30th May, 2015.
Last updated 12th May, 2016.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com