

Domain Modeling

CS 2000: Systems Analysis and
Design

Dave Levitt

Agenda

- Questions on Project / Readings?
- Domain Modeling Concepts
 - Purpose
 - Guidelines
 - Examples
 - Rose Demo (time permitting)
- Lab

Domain Modeling

- A Domain Model is an OOA construct that:
 - Shows abstractions (conceptual classes) in the *problem* domain, I.e. not the *solution* domain.
 - Is best kept at high level so as to verify with end users, and ????
 - Is developed using a mapmaker's strategy.
 - Is built over several iterations in the ?? Phase.
- A Domain Model is drawn like a set of classes, but...
 - Does not show methods.
 - Makes use of conceptual, as opposed to design terms. Example: Audit, not AuditFile.
 - A.k.a. conceptual models, analysis object models.

Domain Modeling (cont'd)

- Strategies for building a domain model:
 - Draw from a list of existing conceptual class categories. See Larman: pages 140 & 141.
 - From nouns in the use cases.
 - From Analysis and Data Modeling Patterns.
 - “Analysis Patterns”, Martin Fowler,
 - “Data Modeling Patterns – Conventions of Thought”, David C. Hay and Richard Barker

Domain Modeling (cont'd)

- Conceptual Class Categories include:
 - Physical objects (credit card swiper)
 - Specifications / Descriptions (song description)
 - Places (terminal, station)
 - Transactions (sale, purchase)
 - Roles (service technician ---beware)
 - Containers (song categories)
 - Things is containers (songs)

Domain Modeling (cont'd)

- Finding Conceptual Classes from noun phrases:
 - Requires well written use cases.
 - Is effective, but watch for ambiguities in the English language. Example: cash drawer. Is this a class or an attribute of one?
 - Can be combined with Conceptual Class Categories.
- Finding conceptual classes is not an exact science.
 - Requires practice, practice, practice.
 - Read books (study other people's models)
 - Accept feedback from peers during reviews.
 - theCoadLetter.com; martinfowler.com

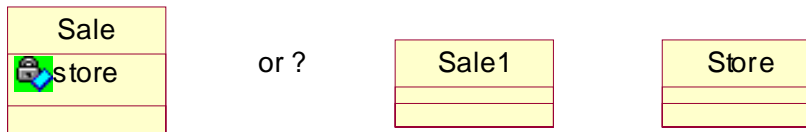
Domain Modeling (cont'd)

- To construct a domain model:
 - Find candidate conceptual classes (see page 143)
 - Draw them in a domain model. (see page 157)
 - Add associations and attributes (next week).
- Domain models trace nicely, but not exactly to design models. This is one of the major benefits of OO.

Domain Modeling (cont'd)

- Common Problem #1: Is a candidate class a class or an attribute of one? (see page 146)
 - If it can't be expressed as a number or text, it's probably a candidate class.
 - If in doubt, make it a class.

Is store an attribute or a class?



Domain Modeling (cont'd)

- Common Problem #2: Assume we are developing a POS system that requires a report of items sold. What happens if an item is deleted? (see page 147)
 - We could say there was one class: item, with attributes (item id, description, price, etc.) but:
 - It's inefficient (description attributes remain the same for all instances of the Item class.
 - It's impossible to report on an item if it is deleted!
 - The solution is to define two conceptual classes – an item (which is the Item ID) and the description of the Item.
 - This is called a specification / description class.
 - Specification classes occur frequently in the analysis model, but they are not appropriate at a use case level. Why?

Domain Modeling (cont'd)

- Common Problem: What level of detail?
 - If you recall, a domain model looks like a class diagram, but the two do not communicate the same thing:
 - At a **conceptual level**, things are described as they would in the real world (or almost). A domain model is a conceptual level construct.
 - At the **specification level**, you are communicating at a design level – but no particular language. Example: class diagram
 - At the **implementation level**, you are communicating at the code level. Example: component diagram, class diagram.
 - There is no right or wrong level, you just have to understand the level you want to communicate at.
 - Note the UML does not care, a class is still a class, it all depends on how you care to communicate / interpret it.

Domain Modeling (con't)

- Class Exercise:
 - Identify candidate classes in the examples
 - Review & Discuss