ISYS2120 sem2 2022

Week 5: Creating a conceptual data model

Alan Fekete

Natural language descriptions

- In most settings, a conceptual model is created by first obtaining descriptions of the domain in natural language (eg English)
  - Ask stakeholders to write about the domain
  - Ask stakeholders to speak and explain
  - Look for existing descriptions (eg induction guides for new staff)
- In classroom situation, this is given to you, in one place! In real-world, you would hunt for as many examples as you can find

Nouns

- Nouns in the text are especially useful
- Consider whether the noun could be an entity type?
  - Are there different instances that can be distinguished from one another?
  - Does each instance have relevant information that the system should keep track of(attributes)?
- Maybe the noun is better as showing that there is an attribute of some entity type?
  - Which entity type would this be providing information about?

Verbs

- Verbs can also be useful. Sometimes they show connections between entities, that can be captured in a relationship type

Constraints

- For every entity set, do you know what identifier can be used to distinguish them (as primary key)?
- For every relationship type, consider cardinality and participation constraints, on each of the sides
  - Text often provides a clue through use of singular or plural, and through use of “may/can”, “must/will”

Clues to inheritance hierarchy

- Look for phrases such as “some X [do something, have something, are also something]”
- Look for “Both C and Y [do something, have something, are something]”
- Look for a significant amount of common information about two entity sets; maybe there is a generalization that makes sense?

Warnings

- Check each attribute you propose, to consider whether it is better seen as the identifier of some other entity set
  - In that case, there should be a relationship set instead, connecting between the two entity sets
- Check each relationship set you propose: if there are many attributes of it, consider if it might instead be “reified” (ie each relationship instance is treated as an object itself) as an entity set, which is then related many-to-one to each of the original participating entities

“Category” entities

- A common situation in manufacturing, inventory etc, is that there can be many objects which are all identical in most aspects
  - Eg library has many copies of a book (edition)
  - Eg supermarket has many tins of a product
  - Eg airline has many planes of a model
- There will usually be an entity set for the category/model/product
- Is there also an entity set for the separate items (individual book copy, tin, plane, etc)?
  - Can they be distinguished [if so, is there a unique identifier across different products, or only within a product category ie weak entity]? Is there any information that needs to track which particular instance is involved?
  - Eg a book copy has a barcode, and is on loan to someone specific or on-shelf
    - But a tin of soup is not distinguished from other tins of the same soup…
    - Discuss: What do you think about a particular plane?
  - If we do not distinguish the copies, maybe just have an attribute of the category, that counts how many copies we have

Fill in gaps

- Look at each entity set and relationship set, to be sure you have attributes for all aspects that are important in the domain
- Look at the ways the system might be used (“business processes”), and check that you have modelled the information needed to deal with each
  - Consider what parameters a user would need to provide, to allow the process to happen

Remove redundancy

- Often, there will be a relationship that can be deduced from other relationships
  - Eg employee works-at office, may be redundant if we know employee works-in department, department located-at office
  - But be careful: if any of the relationships concerned is many-to-many, it may not always be enough to chain them together
- Remove any relationship that is really redundant information

Iterate

- When you have a proposal of a conceptual model, go back through the text and make sure that everything mentioned can be found somewhere from the model, or else you decide explicitly that this is not needed for the system
  - Usually, you will discover a need to extend or modify your model; do so, and then do all the checks again!
- In real-world, show the model to stakeholders and ask them to check it
  - [Unfortunately, this is not always possible for classroom assessment tasks]

Other sources of inspiration

- Existing data sets
  - Paper files, spreadsheets etc
  - These usually will have information that should be attributes in the conceptual model (eg entry on a form or spreadsheet)
  - These may belong as attribute of an entity type you have already recognised, but they may also suggest an extra entity type
- Existing systems or processes
  - Watch staff dealing with situations, and see what information they gather
- [Unfortunately, these are not typically available for classroom assessments!]

Remember

- There is no single ideal conceptual model for a domain
- But aim for something that is useful, and avoids serious flaws

Class case-study (from asst in 2021)

The primary goal of this E-commerce website is to keep track of products, inventory, customers and orders.

Each product has a unique identifier, name, description and maybe some pictures of the product. Each product belongs to a particular product category. Some categories have sub-categories. A product category is identified by its unique identifier and also has a name.

The system should keep track of the inventory levels of products. This includes how much of a particular product is currently in stock and in which location it is stored. As prices change periodically, the system should be able to maintain different listed prices for each product. These prices would have different periods in which they were active and should not overlap.

A user of the system must be either an employee or a customer. Each user should have an id, name, username, password, one or more addresses and one or more contact methods. An address typically has a street number, street name, city, state, postal/zip code, country, unique identifier and usage type.

An employee may also be a customer and they have an employee discount. A customer also has a loyalty number.

A user can make an order of at least one or more products. An order should include information on the date it has been made, a unique identifier, the total price for the order, where it is being shipped to. The system should also keep track of the status of the order along with any dates/times that are associated with a status update.

Class suggestions

- Entity sets
- Relationship sets