Chapter 7

Domain Object Model

Topics

- Contact Management (CM) – the domain
- Domain use case model
- Domain glossary
- Domain class model
Domain Object Model (DOM)

- **Domain**
  - application area (a subsystem) of an enterprise information system
  - defines the scope of an application under development

- **DOM is more than domain model in UP**
  - (class model for the domain)

Use cases and actors

- **Use case** represents a major piece of system functionality

- **Actor**
  - a role that somebody or something plays with regard to a use case
  - communicates with a use case (via «communicate» relationship) and expects from it some feedback – a value or observable result

- **Multiple use case diagrams** (multiple viewpoints) may be desirable:
  - no actors – use cases and their relationships
  - single actor’s viewpoint – all use cases for each important actor

- **Use case diagrams** visualize use cases, actors, and their relationships (but the usefulness of diagrams in a **use case model** are questionable)

- **Use case specifications**, which are text documents stored in a CASE tool repository, are the main power and benefit of use case modelling
Use case relationships

- **generalization** relationship - the execution of one use case enfolded (always) the functionality of the included use case.
- **association** relationship - the execution of one use case may need to be extended (sometimes) by the functionality of the extending use case.

Use case diagram for CM

- **negotiate new customer** relationship - the execution of one use case enfolds (always) the functionality of the included use case.
- **extend** relationship - the execution of one use case may need to be extended (sometimes) by the functionality of the extending use case.
Alternative use case diagram for CM

- Manage Calendar Entries
- Manage Contact Information
- Manage Contact Documents
- Manage Incoming Documents
- Manage Timelog Records
- Manage Outgoing Documents

Domain glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition and Explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td>calendar entry</td>
<td>An appointment, meeting, event or other activity scheduled in an employee’s electronic calendar to act upon. A calendar entry may be timed (when it has due date/time allocated) or it may be untimed (it appears as a “things to do” entry without a specific date/time by which to act upon it).</td>
</tr>
<tr>
<td>contact</td>
<td>A person or organization that AEM communicates or does business with.</td>
</tr>
<tr>
<td>customer</td>
<td>A person or organization that AEM communicates or does business with.</td>
</tr>
<tr>
<td>provider (auxiliary supplier)</td>
<td>A contact that provides auxiliary data or services to AEM. This information includes demographics (divisions of groups of people based on specified attributes such as age, sex, and socio-economic factors), surveys (measured and projected viewing/listening/readership figures per different demographic categories and for different outlets), rates (advertising prices for different outlets to determine the cost of an ad), discounts (estimated reductions in advertising prices due to the purchasing power of an agency when booking ads through a specified medium).</td>
</tr>
<tr>
<td>employee</td>
<td>A person who is employed by AEM or has other work arrangement with AEM with or without pay.</td>
</tr>
</tbody>
</table>
Domain class model

ClassB
attribute1
attribute2
operation1()
operation2()

ClassA

ClassD

ClassC

unidirectional association
association1 0..1
association2 1..n
aggregation
1..n

generalization
1..n

Repository

a database of a visual modeling tool that stores all model specifications, including diagrams
Classes and attributes

- **Class** represents a set of objects
- **Domain classes** signify business entities (hence called also entity classes or "business objects")
- Most domain classes are definitions for multiple **object instances** (hence they are not **singleton classes**)
- **Attribute**
  - represents a data value
  - has a **name** (product_name) and a **data type** (string)
    - data types of attributes are **primitive types**

Class relationships

- **Relationship** is a meaningful connection between classifiers (i.e. classes in the case of a domain class model)
- **Association**
  - specified on classes (types), but represents a relationship between instances of those classes (types)
  - **Multiplicity** specifies how many instances of one class relate to one instance of another class
  - The presence of zero in the multiplicity (0..n or 0..1) indicates that the **participation** of a class instance in the association is optional
- **Aggregation** is a special kind of association where an instance of one class (a whole; superset class) contains instances of another class (a part; subset class)
- **Generalization** is a relationship on classifiers (classes) stating that each instance of the more specific class (subclass) is also an instance of the more general class (superclass)
  - **multiplicity** does not apply to generalization
Class model for CM

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>contact_name, phone, email, status, rating</td>
<td>1:0..n to Organization</td>
</tr>
<tr>
<td>Event</td>
<td>description, date, time</td>
<td>0..n to Contact, Task</td>
</tr>
<tr>
<td>Task</td>
<td>purpose, value</td>
<td></td>
</tr>
<tr>
<td>Employee</td>
<td>emp_id, emp_name</td>
<td>0..n to Contact, Event</td>
</tr>
<tr>
<td>Document</td>
<td>doc_subject, content, date</td>
<td>0..n to Contact, TimeLog, Calendar</td>
</tr>
<tr>
<td>TimeLog</td>
<td>record_desc, date, time</td>
<td>0..n to Employee, Calendar</td>
</tr>
<tr>
<td>Calendar</td>
<td>entry_desc, date, time, status</td>
<td>0..n to Employee, Task</td>
</tr>
<tr>
<td>CustomerReqs</td>
<td>report_category, market_range, media_range, product_range, advertiser_range</td>
<td>0..n to Contact</td>
</tr>
</tbody>
</table>

Alternative class model for CM

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes</th>
<th>Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>contact_name, phone, email, status, rating</td>
<td>1:0..n to Organization</td>
</tr>
<tr>
<td>Event</td>
<td>description, date, time</td>
<td>0..n to Contact, Task</td>
</tr>
<tr>
<td>Task</td>
<td>purpose, value</td>
<td></td>
</tr>
<tr>
<td>Employee</td>
<td>emp_id, emp_name</td>
<td>0..n to Contact, Event</td>
</tr>
<tr>
<td>Document</td>
<td>doc_subject, content, date</td>
<td>0..n to Contact, TimeLog, Calendar</td>
</tr>
<tr>
<td>TimeLog</td>
<td>record_desc, date, time</td>
<td>0..n to Employee, Calendar</td>
</tr>
<tr>
<td>Calendar</td>
<td>entry_desc, date, time, status</td>
<td>0..n to Employee, Task</td>
</tr>
<tr>
<td>CustomerReqs</td>
<td>report_category, market_range, media_range, product_range, advertiser_range</td>
<td>0..n to Contact</td>
</tr>
</tbody>
</table>
Summary

- **Domain Object Model (DOM)** is a model of one application area (domain) of an enterprise.
- **Domain use case model** is a UML use case diagram and associated use case documents defined for a domain.
- **Domain glossary** expands a business glossary by adding to it terms and definitions specific to the domain.
- **Domain class model** defines classes within a domain and relationships between these classes:
  - There are three main kinds of relationships: **associations**, **aggregations**, and **generalizations**.