Domain analysis

Remember the stages of the software lifecycle

- Domain analysis
  - *The wider business context for the system*
- Requirements
- Specification
- Architecture
- …

If software spends most of its time being changed and maintained, it's important to understand the context in which it lives

- Changes will move the application around this contextual space
- Need to understand where it might go
- Changes are determined by the business context

What is a "Domain"?

Berard gives two characterisations:

1. A collection of current and future (software) applications that share a set of common characteristics
2. A well-defined set of characteristics that accurately, narrowly, and completely describe a family of problems for which computer application solutions are being, and will be sought

A Domain Expert

An individual who is both experienced and knowledgeable about a particular application domain

He or she must have detailed knowledge about available COTS products and the interface standards they adhere to

There must be at least one domain expert for each application domain

A Domain Analyst

He or she is responsible for the development of the appropriate domain analysis classification scheme and the criteria for selection of potentially reusable components

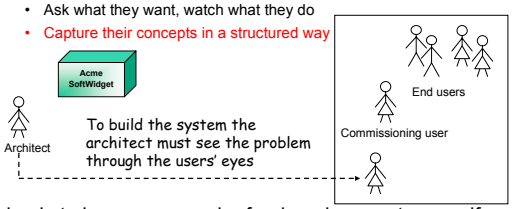
Determining opportunities for the composition of components into higher-level structures is also an important part of the domain analysis process

The domain analyst will interact with the domain expert as part of this process

Domain analysis tasks

Conversations with *domain experts*

- Ask what they want, watch what they do
- Capture their concepts in a structured way



Idea is to become enough of a domain expert yourself that you understand what the users are saying

- Can you pass for a travel agent after analysing a travel system?

Domain analysis tasks

Characterise and understand the problem space

- To factor out commonalties

Characterise and understand the solution space

Both looking at what it is now and what it should be after factoring commonalties

Create a model of the Domain

- Domain Engineering extends the domain analysis to include the actual design and construction of the new solution space

Domain Classification

These Classifications can help form the basis for developing an organisational reuse strategy

Domain Classification

- Domain-independent software: graphical user interface
- Domain-specific: invoicing a customer for goods ordered
- Application-specific: control functions of a specific type of washing machine

Domain Independent Software

Graphical User Interface Functions, Math Libraries, Abstract data types

- This accounts for about 20% of an application as a rule of thumb [Poulin]
- Efforts at Reuse of software can hope to reduce development effort by 20%

Horizontal Reuse - Cuts across several Domains

Domain Specific Software

Navigation and Control for Aircraft, Flight Control, Attitude and control

Database Management Systems, Spreadsheet Software, Word-processing, Desktop Publishing

- Can account for up to 80% of the code [Poulin]
- Efforts at Reuse of software from this category can hope to reduce development effort by up to 80%

Called Vertical Reuse - Reuse is within the domain only

Largest payoff points to concentration on Vertical Reuse

Application Specific Software

Application Specific Software

- Handles the unique details of a customers requirements specification
- Custom Code
- Typically accounts for about 15% of an application [Poulin]

Little potential for reuse of Custom Code

Domain modelling

Once you've finished domain analysis you need to create a model of the Domain

The idea is to capture the business and understand it

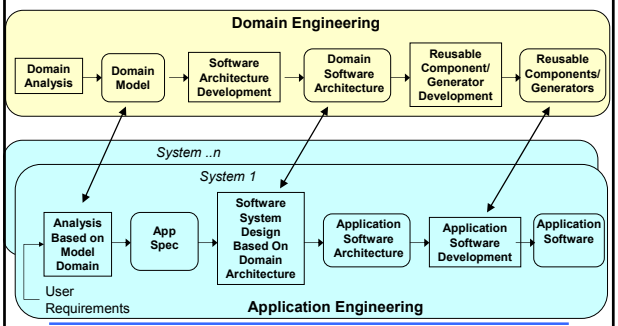
Don't just concentrate on *this* application, study *all* applications in this business first

- That's where the software has to live and work
- …and the context that will exert pressure on it to change

In an object-oriented world, we capture the objects, relationships and processes in the business

- The people and roles in the organisation
- The tasks they need to perform, and the interactions between these tasks
- Good documentation for the business

Domain engineering and application engineering



OO analysis and domain analysis

While OO analysis is focused on the features and functionality of a single system to be generated, a domain analysis focuses on the common and variant features across a family of systems

Use cases and sequence diagrams are good for both OO and domain models

- The actors and their tasks
- The sequences of actions and states

Class diagrams are good for OO models

- The "things" in the system, through the users' eyes

Wrapping up

Domain analysis focuses on problems, not solutions

Domain analysis is about understanding the context for the application

- Understand all the applications, not just this one
- Talk to domain experts and watch them at work
- Think like the customer as far as possible

Domain models, software architectures and re-usable components are artefacts of domain engineering

Bibliography

E. Berard. *Essays in Object-Oriented Software Engineering*. Prentice Hall, 1992.

J. S. Poulin. *Measuring software reuse : principles, practices, and economic models*. 1997.

Tutorial

UML and Rational Rose

Rational Rose: CASE tool for UML
Includes an editor to draw UML diagrams, code generator, ..

http://www.dsg.cs.tcd.ie/~meierr

Rational Rose is available in the ICT labs
Will be required for your course assignment