L2 - Introducing inception

CS3028 - Principles of Software Engineering

Ernesto Compatangelo

Department of Computing Science

UNIVERSITY OF ABERDEEN

2.1 Reminding past issues and mapping them to current topics

Where are we now?

Software development paradigms

⇒ The Unified Process (UP) paradigm

⇒ UP phases and UP disciplines (activities) within each phase

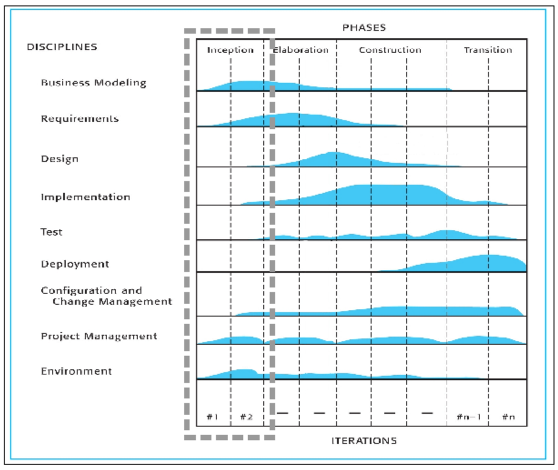
⇒ Inception (first UP phase)

⇒ Activities and deliverables during inception

⇒ Inception concepts and (UML) notations

What is inception?

The early stage of the UP software development paradigm, focusing on:



- **Business modelling** — understand the domain
- **Requirements** — make business case; state vision; draft top 'use cases'
- **Design** — draft software architecture
- **Implementation** — Prototype critical features

2.2 Activities (disciplines) during inception

Business modelling during inception

The objective of the UP business modelling discipline in this phase is to

- Identify the **business elements** relevant to specify the intended software system & model them as **domain 'classes' and 'objects'** (*e.g.*, using *UML class and object diagrams*)
- Establish **conceptual relationships** between business elements (*e.g.*, using **'associations'** in UML class and object diagrams)
- Identify & model the **workflow** of business elements & process control in the application domain (*e.g.*, using *UML activity diagram*)

2.3 Justifying business modelling

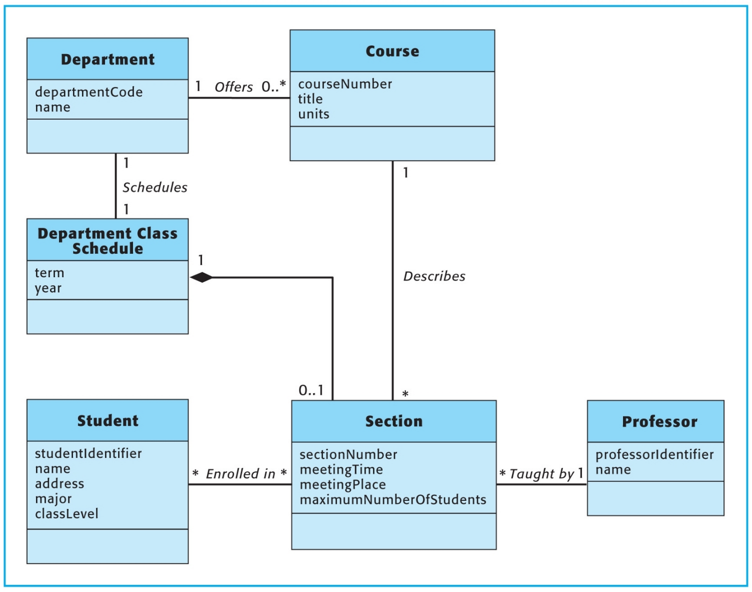
- The organisation and its current problems must be clearly understood before (at least partial) automation is considered
- Automating a business changes (part of) both
  its organisational structure and its workflows
- Processes & information may | may not add value to an organisation:
  in the first case they should be automated;
  in the second case, they should be eliminated
- Modelling and improving a business area and its processes *may* thus precede the definition of users' requirements for a software system

2.4 Modelling domain elements (a.k.a. conceptual modelling)

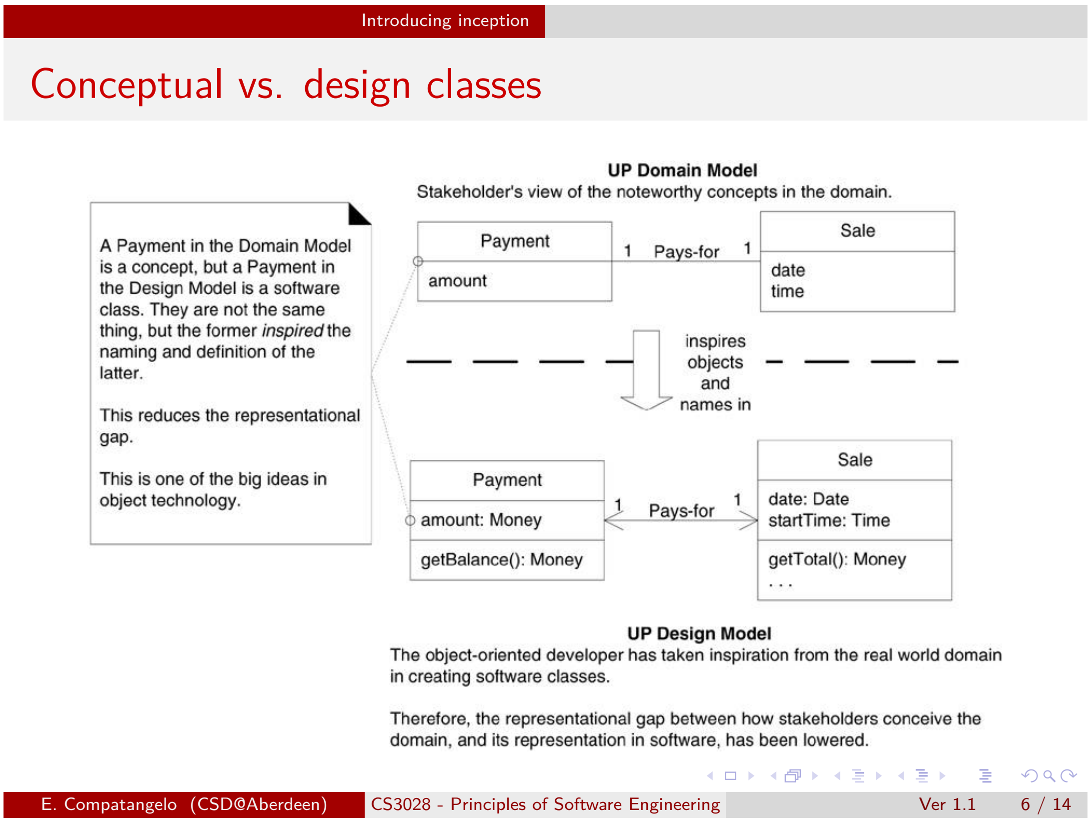
- A representation of conceptual classes or real-world domain objects
- **NOT** a set of diagrams describing software classes
- IN UML notations, a set of class diagrams in which no operations (method signatures) are defined.
- A conceptual perspective which may show:
  - domain objects or conceptual classes
  - associations between conceptual classes
  - attributes of conceptual classes

Introducing inception

Business level UML class diagram example



E. Compatangelo (CSD@Aberdeen) CS3028 - Principles of Software Engineering Ver 1.1 5 / 14



Introducing inception

Conceptual vs. design classes

UP Domain Model
Stakeholder's view of the noteworthy concepts in the domain.

A Payment in the Domain Model is a concept, but a Payment in the Design Model is a software class. They are not the same thing, but the former *inspired* the naming and definition of the latter.

This reduces the representational gap.

This is one of the big ideas in object technology.

UP Design Model
The object-oriented developer has taken inspiration from the real world domain in creating software classes.

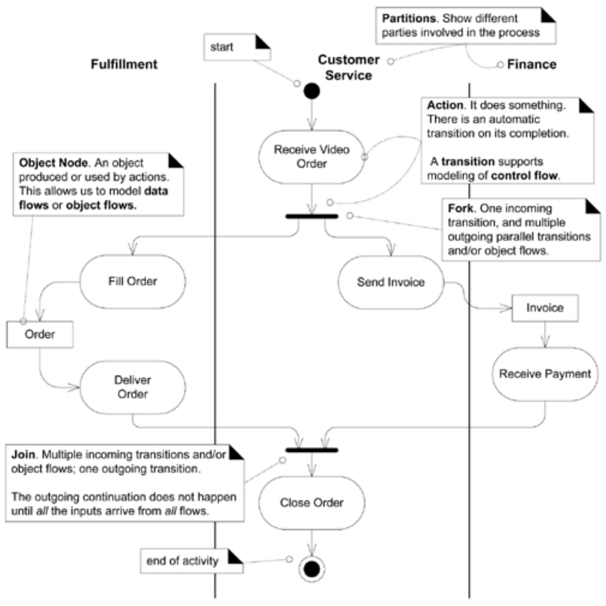
Therefore, the representational gap between how stakeholders conceive the domain, and its representation in software, has been lowered.

E. Compatangelo (CSD@Aberdeen) CS3028 - Principles of Software Engineering Ver 1.1 6 / 14

2.5 Modelling workflows (a.k.a. activity modelling)

- A workflow is a combination of sequences, iterations, and selections of possible ‘elementary activities’ (or actions)
- A workflow may involve a (combination of) control flow, information flow, and material flow
- UML activity diagrams can be used to describe activities in a human organisation involving control, information, and material flows
- Activity diagrams at this stage:
  1. **NEITHER** describe strict data and control flow in software systems
  2. **NOR** describe algorithmic procedurality in a method

Business level UML activity diagram example


Partitions. Show different parties involved in the process
start
Fulfillment
Customer Service
Finance
Action. It does something. There is an automatic transition on its completion.
A transition supports modeling of control flow.
Receive Video Order
Object Node. An object produced or used by actions. This allows us to model data flows or object flows.
Fork. One incoming transition, and multiple outgoing parallel transitions and/or object flows.
Fill Order
Send Invoice
Invoice
Order
Receive Payment
Deliver Order
Join. Multiple incoming transitions and/or object flows; one outgoing transition.
The outgoing continuation does not happen until all the inputs arrive from all flows.
Close Order
end of activity

Requirements during inception

The objective of the UP requirements discipline in this phase is to

1. Outline a **vision** for the software product.

   This means identifying and stating goals, expectations, key features, and constraints for the envisaged product.
2. Develop an initial **business case**.

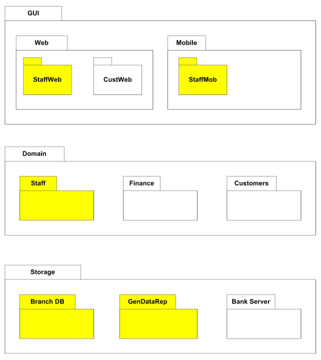
   This means assessing the business value of the SW product, analysing its economical, technical, and organisational feasibility.
3. Provide an initial **requirements specification**.

   This means describing the top-level functional and non-functional requirements of the SW product, including a GUI mockup.

Design during inception

The objective of the UP design discipline in this phase is to

1. Draft a system decomposition
   using layering and partitioning ‘architectural styles’
2. Sketch a top-level system architecture
   that gathers layers and partitions into ‘subsystems’



Implementation during inception

The objective of the UP implementation discipline in this phase is to

1. Avoid to reinvent the wheel — identify suitable off-the-shelf components (*e.g.*, java libraries) that can be used to provide (part of) the envisaged functionalities
2. Evaluate different competing alternatives if any
3. Perform exploratory coding ‘in-the-small’ of critical parts, analysing serialisation, links to databases, client-server connections. . .

Project management during inception

The objective of the UP project management discipline in this phase is to

1. Devise an initial project plan, establishing tasks, deadlines, and deliverables
2. Allocate these to project members after having assigned a role and responsibilities to each of them.
3. Perform an initial assessment of all possible risks associated to the project
4. Draft contingency plans for each potentially risky situation.

Environment during inception

The objective of the UP environment discipline in this phase is to

1. Identify the most suitable Integrated Development Environments for project development (*e.g.*, Netbeans, Eclipse), evaluate and compare them, choose one, and stick to it for the whole project duration
2. Identify further CASE tools that could be used for project development, (*e.g.*, Visual Paradigm, Borland Together, Netbeans UML plugin) evaluate and compare them, choose one, and stick to it for the whole project duration
3. Identify the most suitable versioning tools that should be used for cooperative software development (*e.g.*, Git, Subversion, TortoiseSVN, WinMerge, Netbeans/Eclipse Git/SVN plugins) and stick to them for the whole project duration

Artifacts resulting from the UP inception phase

1. **Vision document** *(Req.)*: summarises key features and major constraints for the system (*i.e.*, its *top-level requirements*).
2. **Initial business case** *(Req.)*: describes business context, defining measurable business and financial criteria for project success.
3. **Critical use case model** *(Req.)*: describes critical functionalities for system acceptance, driving design decisions.
4. **Candidate system architecture** (Des): describes significant decisions about the software components and their organisation.
5. **Initial risk assessment** *(Proj. mgmt)*: identifies and prioritises the most significant obstacles to the successful project completion
6. **Initial project plan** *(Proj. mgmt)*: describes the sequence in which system functionalities will be specified, designed, and constructed. Includes schedule, milestones, iterations for each phase, and budget.

All these in Deliverable 1!

2.6 Preparing for the topics ahead

Next week. . .

Inception disciplines

More specifically, we will focus on:

- Requirements (what meaningful product features?)
- Design (how product features map into software components)
- Software project management (how to manage the development process)