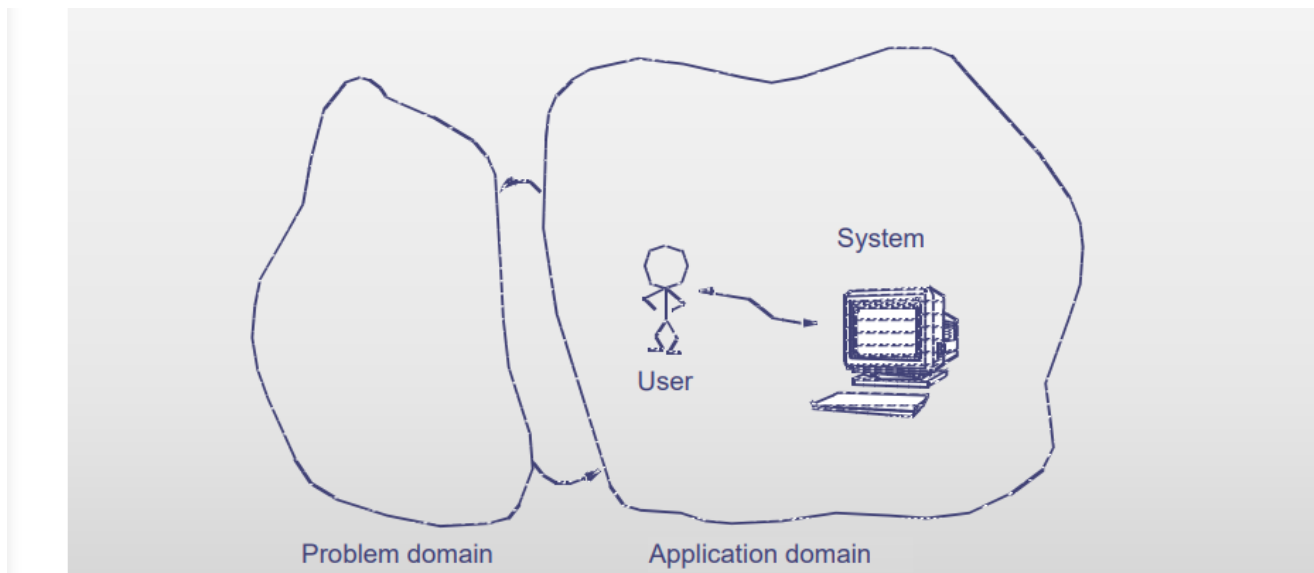


1: Method: Base understanding and analysis.



System = Collection of components that implement modeling requirements, functions and interface.



Interface (UI)	Connection to users or equipment. Text, graphics, printouts, etc. Rader and sensors
Functions	Contains the facilities through the interface(users/equipment) updates the model
Model	Contains dynamic model of the system's problem domain

- Model the context: Model both app. and problem domain during A&D.
- Emphasize the architecture: Aim for easy-to-understand architecture.
- Reuse patterns: use common patterns for quality and productivity.
- Tailor the method to suit specific projects: OOA&D has to be shaped to fit the project.

Analysis

- Physical space: where is the system gonna get used? Mobil in a classroom or somewhere else?
- System requirements

Design

- Produce a system design without significant uncertainties

Implementation

- Realize a design on a technical platform

Method approaches:

- Waterfall: analysis for all -> Design for all -> (sequential)
- Iterative: analysis, design, implementation for a part -> next part -> ...

Object

- Identity: myChair
- State: by dinning table, free, yellow ... etc.
- Behaviour: bought, moved to.., sat down on, ... etc. (EVENTS)

Class

- Structure: has an owner.
- Attributes: position, vacant.
- Behavioural pattern: buy + {move | sit down on + get up from} + sell.

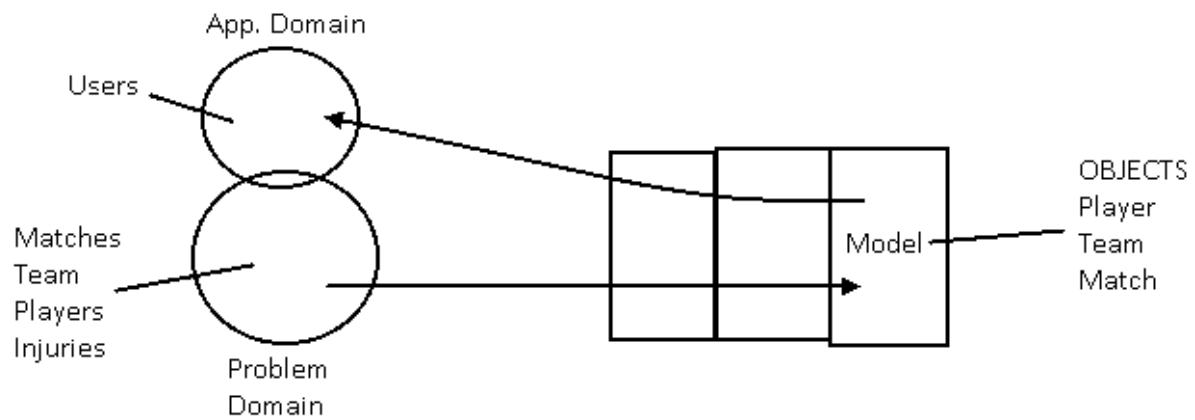
Object examples

Warehouse	Gravel pit
Pallet	Tools
Goods	Sorting machine
Employee	Pile
Truck	Grain
Box	Materiel
Manager	Rocks (different sizes)

Objects in Analysis and Design

Analysis	Design (and implementation)
Outside computer system.	Inside computer system.
Identity: identifies object.	Identity: get access to an object.
Behaviour: events objects has performed or suffered.	Behaviour: an operation an object can perform.

Sports Website: System



2: System Choice: Rich Picture, FACTOR.

Rich Picture: a drawing to illustrate of the problem. to differ problem domain and app. domain. Rich picture = both system and real world.

Factor

- **Functionality:** The system functions that support the app-domain tasks.
- **App-domain:** Those parts of an organization that administrate, monitor or control a problem domain.
- **Conditions:** The conditions under which the system will be developed and used.
- **Technology:** Both the technology used to develop the system and technology on which the system will run.
- **Objects:** The main objects in the problem domain.
- **Responsibility:** The systems overall responsibility in relation to its context.