



Tendiendo Puentes entre la IPO – IS
Madrid, Diciembre 2005

Diseño de Interfaces de Usuario Basado en Modelos

Oscar Pastor

Department of Information Systems and Computation
Valencia University of Technology, Spain



About the central problem...

- How to include interaction modeling in conventional conceptual modeling?
- What kind of tools should the community provide to help to solve the problem?

Delivering IT to the business

IT Development offering

1. We may increase costs at any stage
2. We may or may not deliver on time
3. We might deliver the following functionality
της φIRST φυνχτιον
σομετηινγ ελσε ωε μιγητ δελιτωερ
ανδ περηαπσ τηισ τοο
4. We do not guarantee the system will work
5. You will be blamed for any problems

I accept these terms

Signed _____
A Businessperson

Source: Prof. Tony Morgan, Northface University, UTAH, USA, Doing IT Better, 2005

3

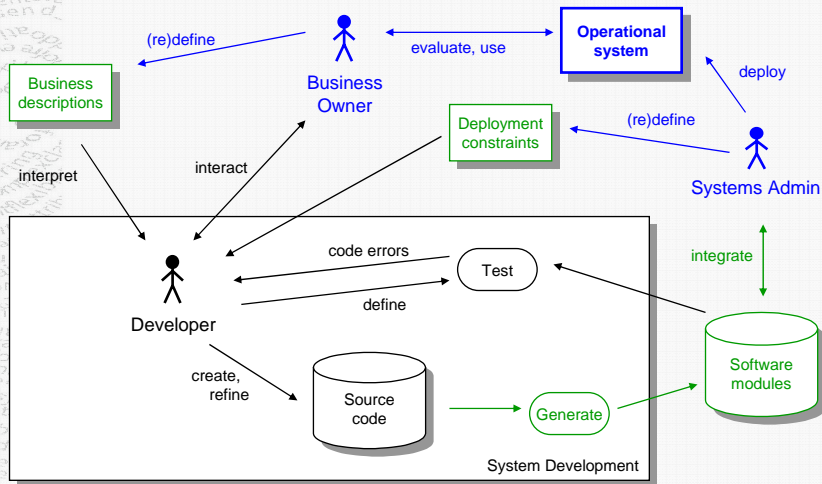
The issue

- Producing an Information System today is:
 - costly (expensive resources over extended periods)
 - much too slow for modern business conditions
 - very risky (hard to control, high failure rate)
 - highly unsafe (introduces hidden failure points)
- The development process has not changed much over the past 40 years
- Is it worth looking for a better way ?

Source: Prof. Tony Morgan, NU, UTAH, USA, Doing IT Better, 2005

4

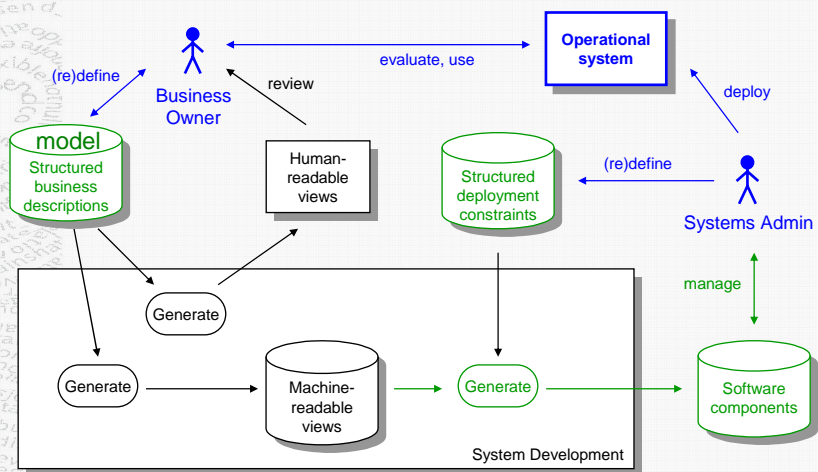
Extreme Programming (XP)



Source: Prof. Tony Morgan, NU, UTAH, USA, Doing IT Better, 2005

5

Extreme Non-Programming (XNP)



Source: Prof. Tony Morgan, NU, UTAH, USA, Doing IT Better, 2005

6

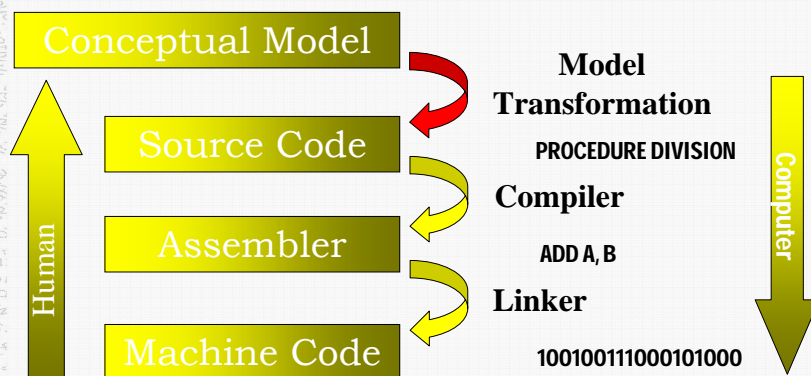
CSCD in a Nutshell

*“To develop an information system
it is necessary and sufficient
to define its conceptual schema”*

Source: Prof. Antoni Olivé, CAISE 2005 Keynote

7

A new Paradigm: Model Execution



8

Some final reflections...

Adoption requires tools!!!

Why haven't we sound tools accompanying these initiatives?

Usefulness should be obvious under the so bad current situation

Usability should be just a logical consequence of adoption and usefulness

9



Two main ideas...

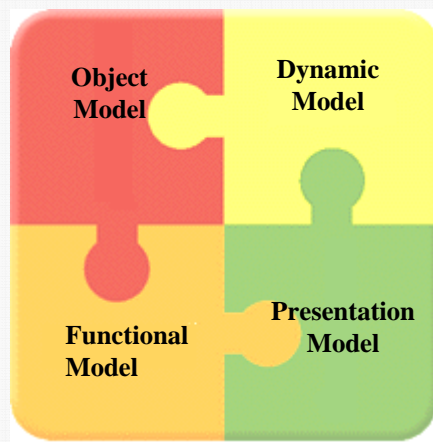
“The history of software engineering has been one of growing levels of abstraction”

“Behind any programming decision there is always a concept that we can capture in a Conceptual Schema”

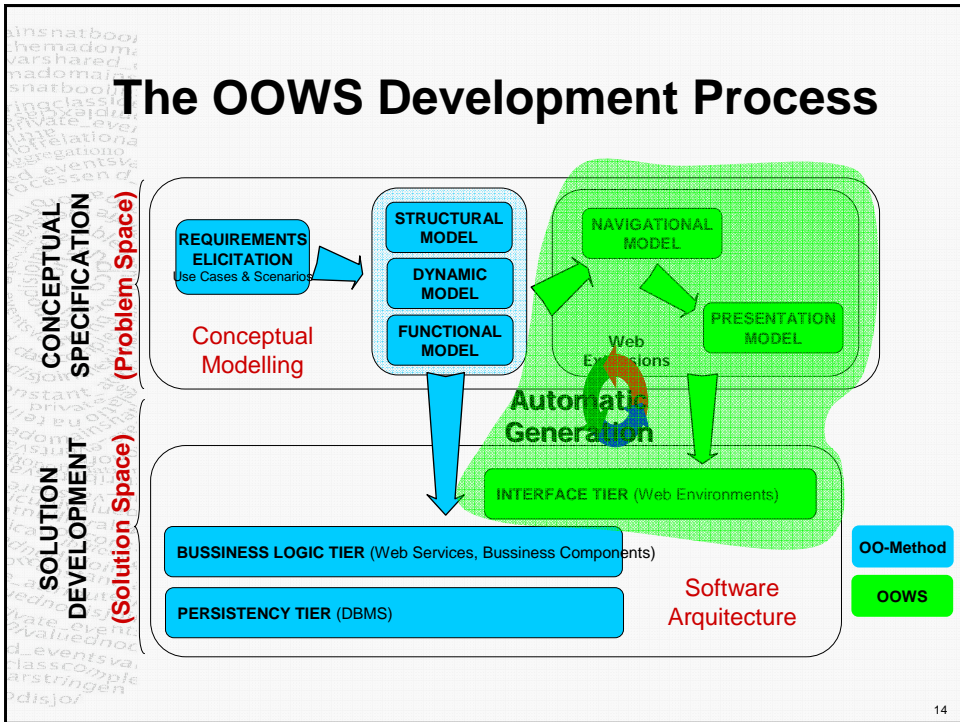
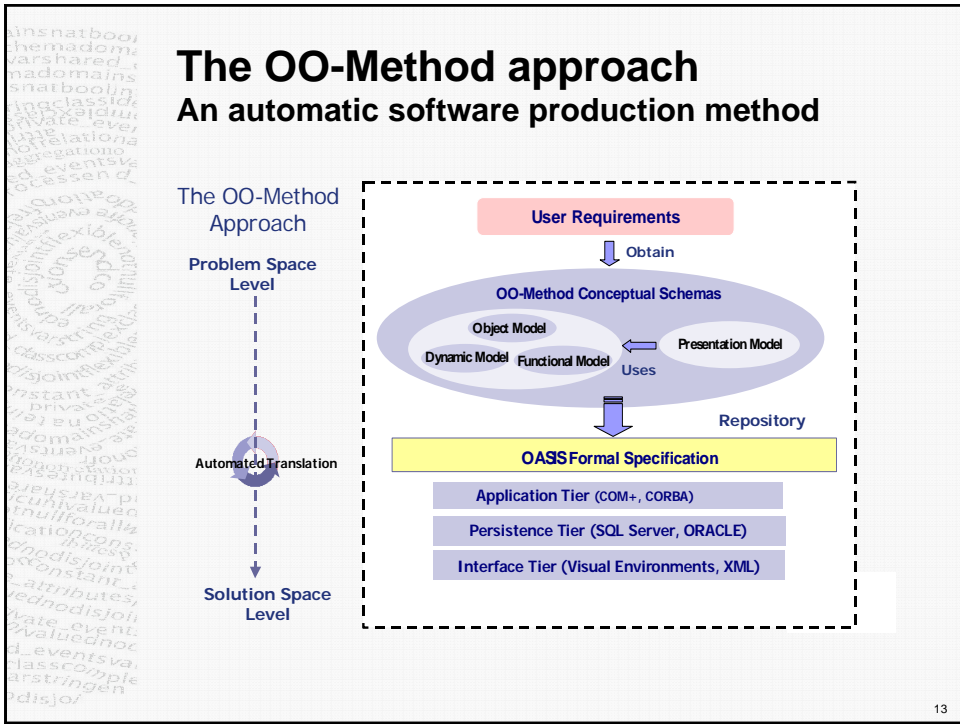
11

Conceptual Model

It contains all functional requirements



12



Navigational Model: Example

Data about member

Related data of the member

Members

Entity

RGroup

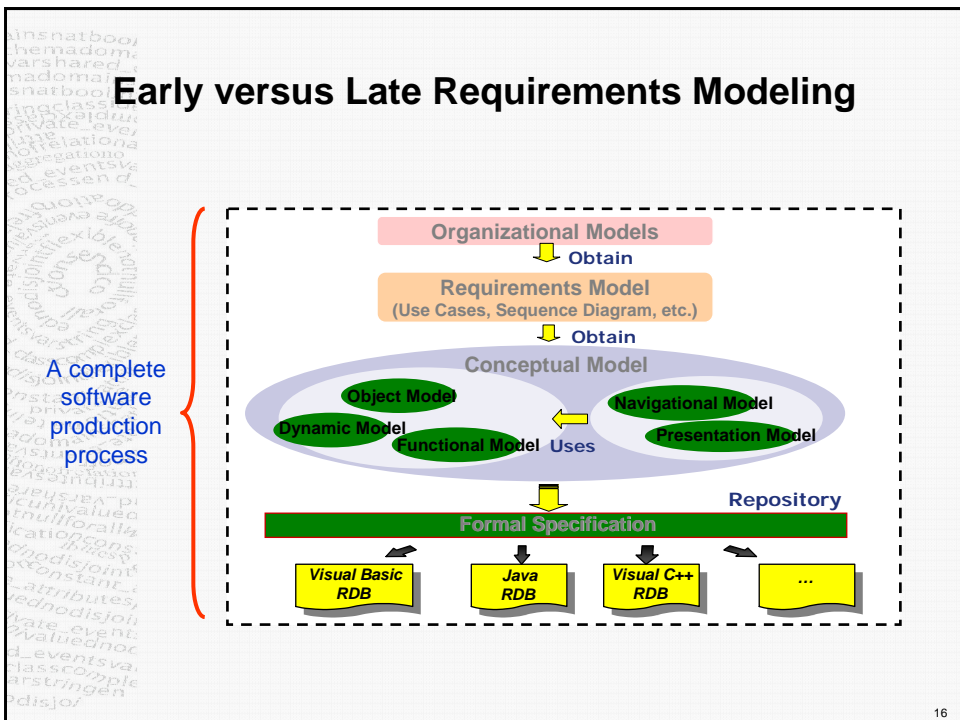
ATTRIBUTE ACCESS STRUCTURE Members
 ATTRIBUTES name, email, isPhD, WorkOn.status
 LINK ATTRIBUTES name

ATTRIBUTE FILTER ByName
 ATTRIBUTES name
 TYPE APPROXIMATE

This web page provides information about MEMBERS of a research group

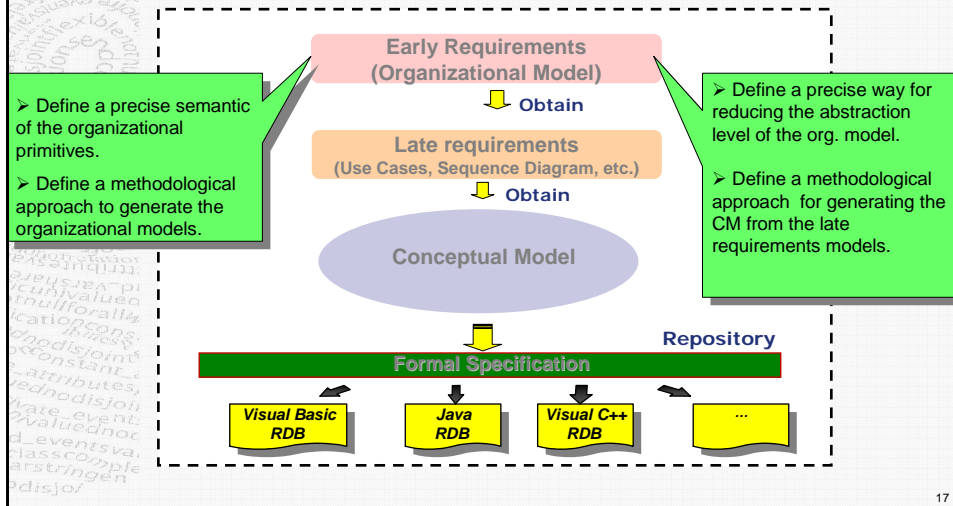
Modelled in OOWS as a Navigational Context

Early versus Late Requirements Modeling



Early versus Late Requirements Modeling

The current state of the OO-Method approach



17



Conceptual Model

*Formal details are **abstracted** by using graphical models with **standard** notation*

- Object Model
 - Class Configuration Diagram
- Dynamic Model
 - State Transition Diagram
 - Object Interaction Diagram
- Functional Model
- Presentation Model



18

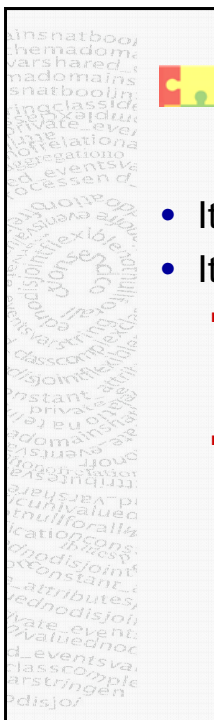




Object Model

Specification of Static properties

- Defined by a **Class Configuration Diagram**.
 - **Classes**
 - Attributes
 - Preconditions and Services
 - Integrity Constraints
 - **Relationships between classes**
 - Aggregation
 - Inheritance
 - **Agents**

19

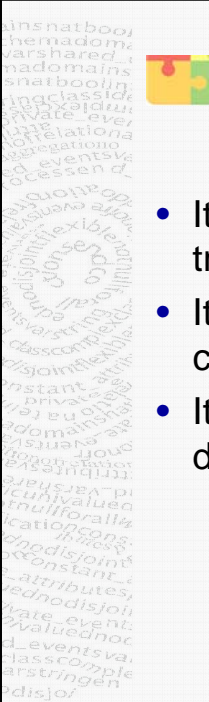




Dynamic Model

Specification of Dynamic properties

- It represents the system behavior
- It is defined by two diagrams:
 - **State Transition Diagram (STD)**
 - It specifies the Object's valid life
 - It defines the available services in each state.
 - **Object Interaction Diagram (OID)**
 - It specifies valid Interaction between objects such as Global Transactions and Operations and Triggers.

20

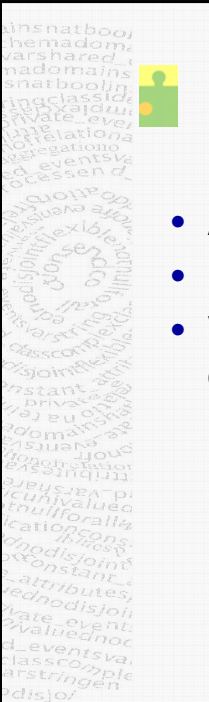




Functional Model

Specifies Static - Dynamic relationship

- It defines the semantics related to state transitions
- It describes how the execution of events changes the value of class attributes
- It is defined by means of a dialog (not a diagram)

21

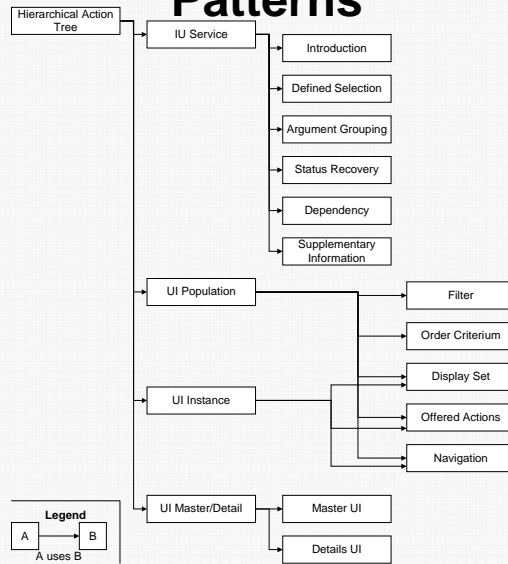
Presentation Model

Specification of the User Interface requirements

- Abstract specification of the user interface
- Independent of the target device
- View Oriented: Language of patterns organized in three layers

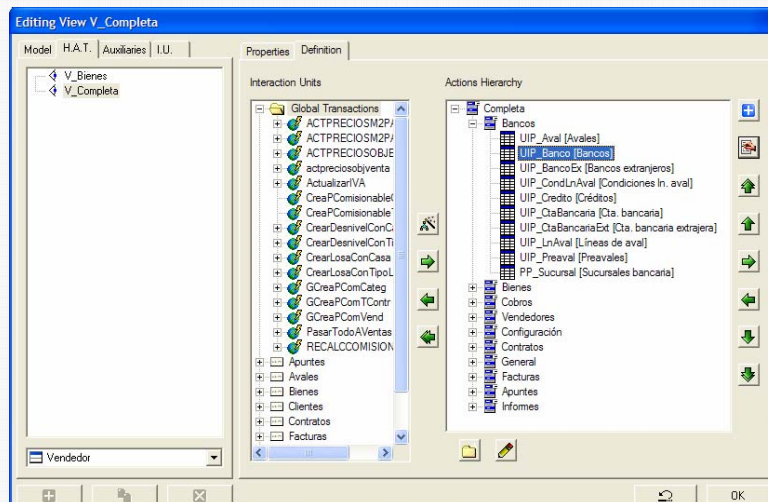
22

17 Existing Presentation Patterns



23

Presentation Model



24

How to embed the PM in the software production process

- Current flow in ONME

