

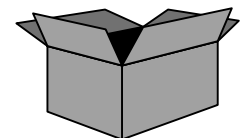
Hybrid Development Methodology for Client-Server Applications

Václav ěpa

Prague University of Economics

Czech Republic

BIT 95



Hybrid Development Methodology for Client-Server Applications

Structure of presentation

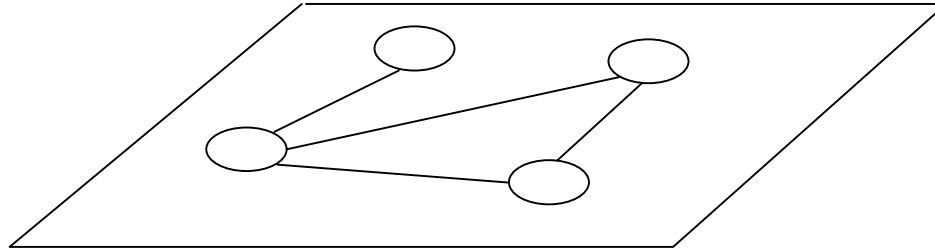
- What is the PDIT methodology
- Underlying principles
- Main features of the PDIT
- Procedure of analysis in the PDIT
- Model of reality - central part of the analysis

PDIT - IS Development Methodology

- target environment for the IS:
 - relational database system
 - client-server DB technology
- development environment:
 - CASE tool supporting the Structured Analysis and Design
 - GUI oriented 4GL

Three Architectures

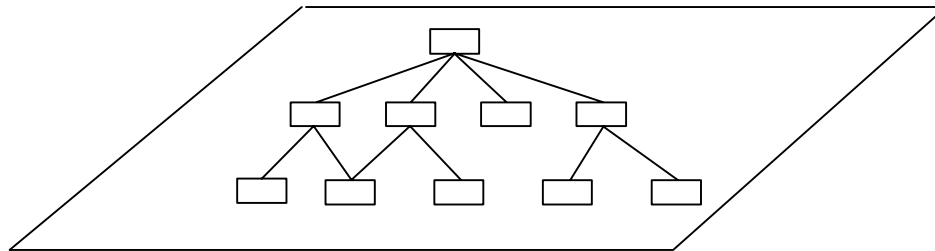
Model of reality



Conceptual
level

Design Techniques and Tools

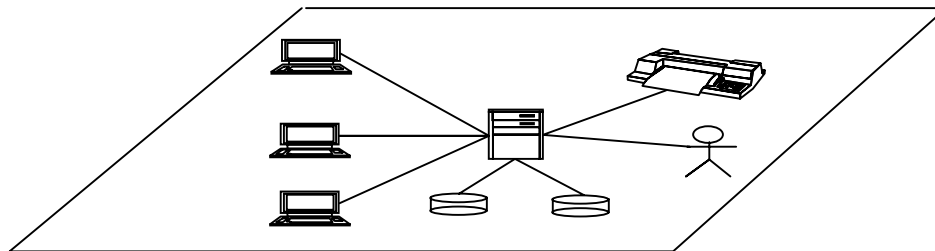
Technological
model



Technology
level

Implementation Techniques
and Tools

Implementation
model



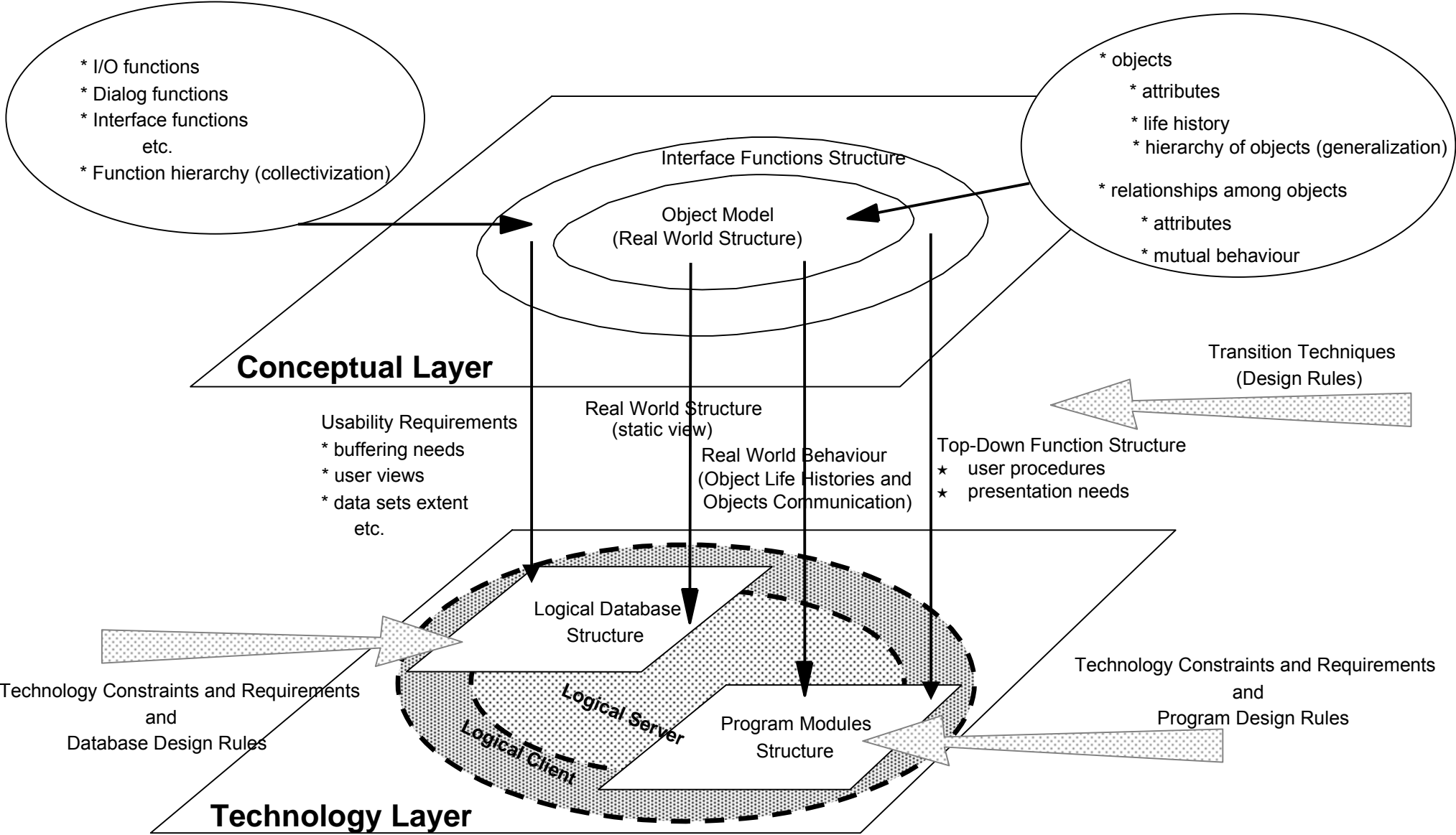
Physical
level

PDIT - IS Development Methodology

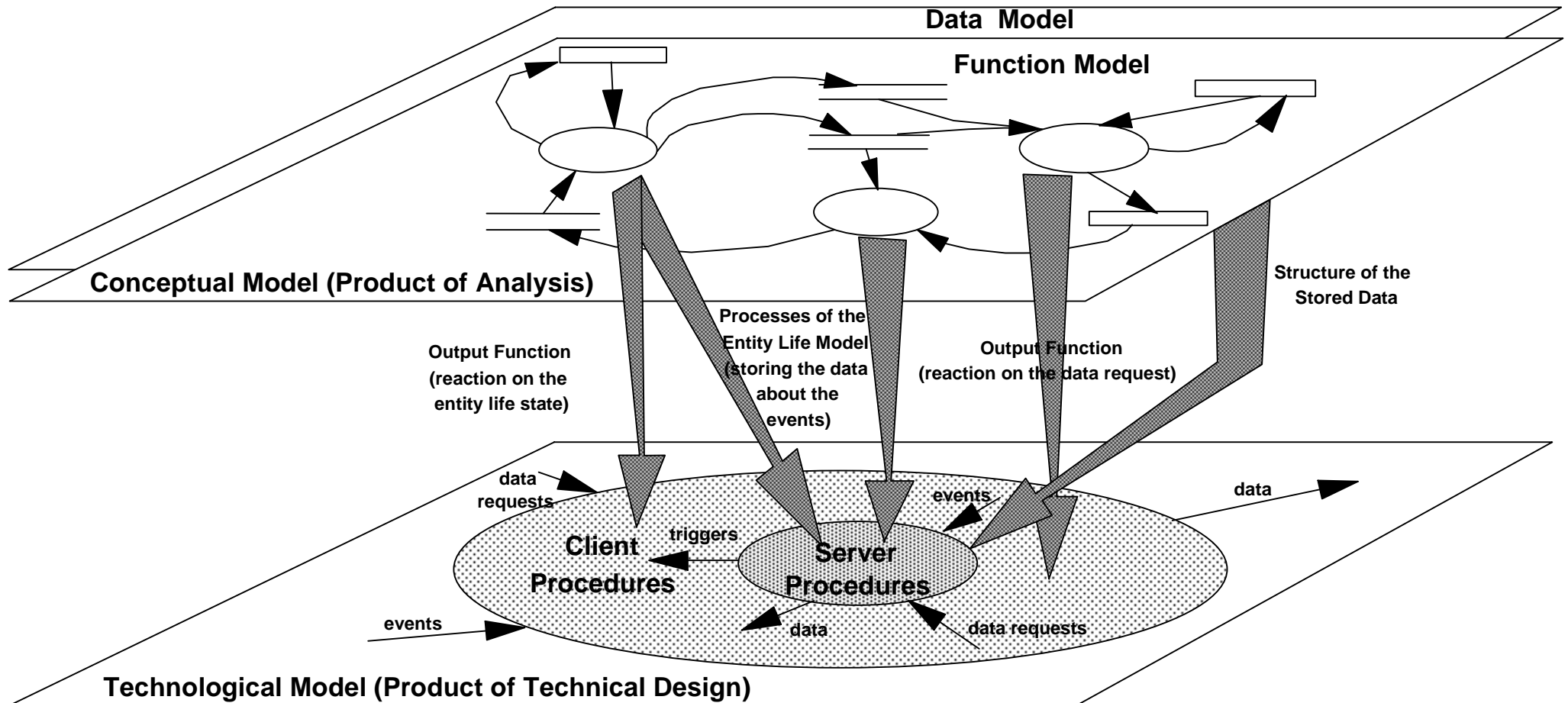
Main Features

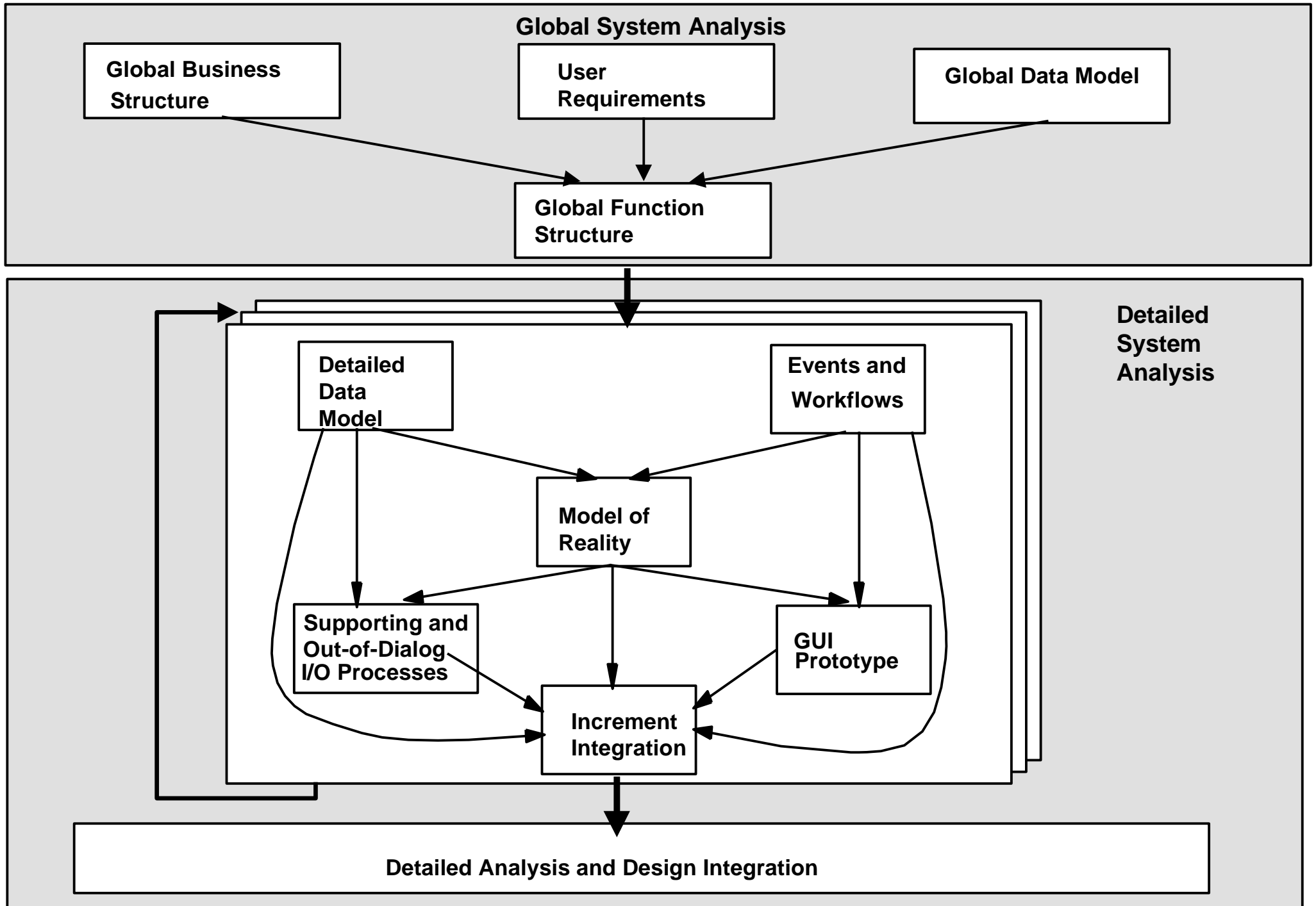
- **Hybrid Methodology** (structured as well as OO)
 - OO nature of the CS technology
 - Structured nature of the Global Analysis and
 - Separation of the data and processes in the target environment
- **Prototyping** as the natural approach for the 4GL tools using
- **Conceptual origin of the Client-Server partitioning**

Transition From the Conceptual to the Technological Architecture



Relations between Conceptual Design and Client-Server Partitioning





Model of reality

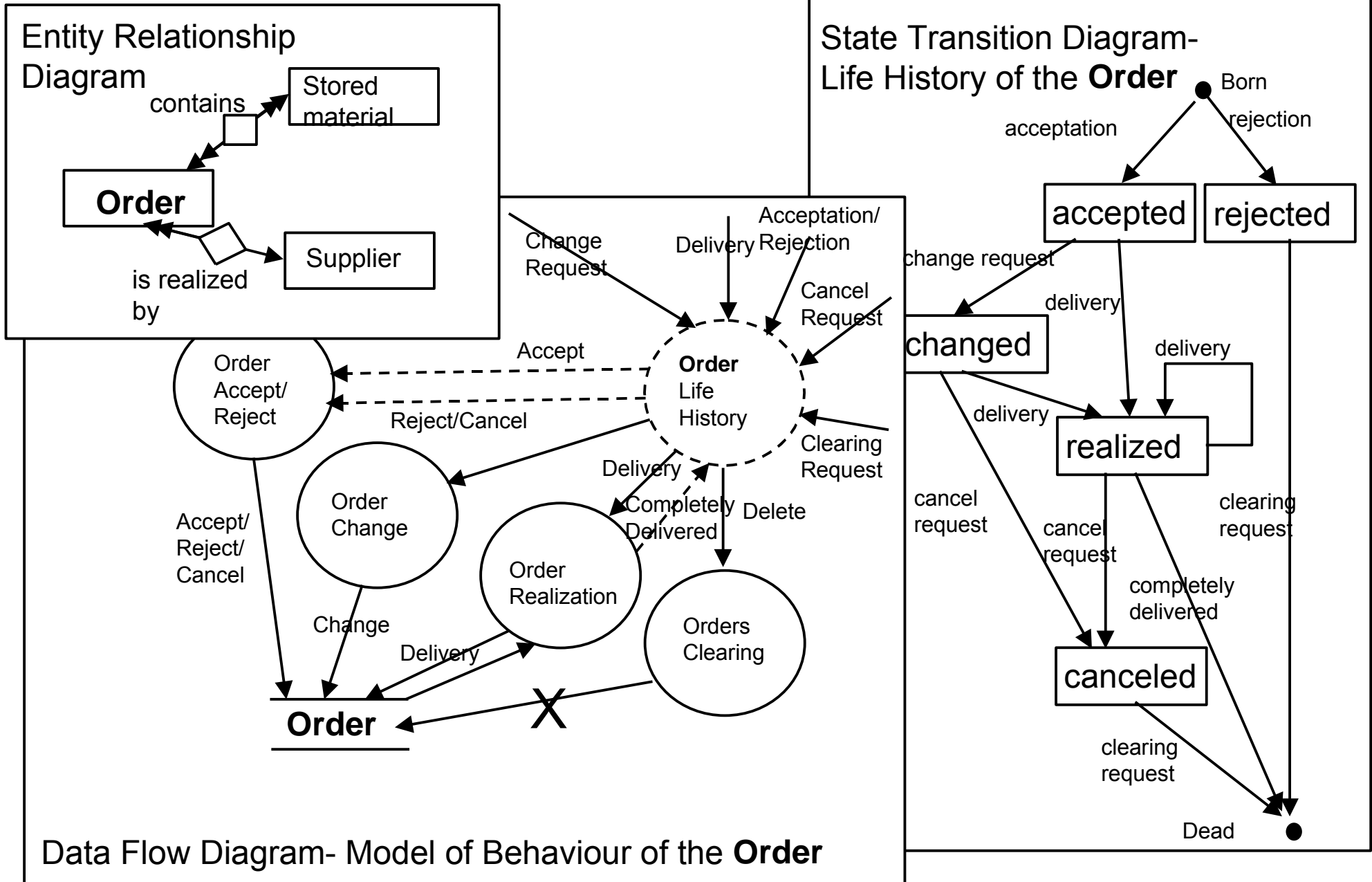
- a set of Data Flow Diagrams (DFD) and
- an Entity Relationship Diagram (ERD)

Entity Behaviour Model

describes the behaviour of one object (entity or relationship) from the Data Model (ERD).

- **entity** (or relationship in the sense of associative entity) from the Data Model (data part of the object)
- **several essential functions** (actions of object state changes)
- **one control process** (STD) (object life history)
- **control flows** between the STD and essential functions (events and actions of the object).
 - **input** control flow (event stimulating the transition of the object from one state to the other one)
 - **output** control flow (action of the object as the reaction on the event)
- **data flows**
 - **from the input/output interface** to the essential function of the Entity Behaviour Model. (data part of the real-world event)
 - **from (or to) the essential function of the other object** (communication of the Entity Behaviour Models - transition of the information about real-world events between objects)

Entity Behaviour Model



Mapping of the Conceptual Model to the Client-Server 3-tier Architecture

