1st DBTech Pro WorkShop

The ACID Principle

Thessaloniki

27-28. November 2003

Transaction integrity

- Discussing **short transactions**
  - always a black box in user's point of view
  - typical in commerce applications
  - **not** discussing long-lived transactions, i.e. duration of days, weeks, months,…
    - users are aware of the inner details of long TX
    - e.g. workflow applicatios, design project app,…
- Take care of LUW's (logical units of work) in applications
- Database is expected to be recoverable

The key

- Transactions are key - even to Structuring Distributed Applications
- ACID properties ease exception handling
  - **Atomic**: all or nothing
  - **Consistent**: transformation to a permitted state
  - **Isolated**: no concurrency anomalies
  - **Durable**: committed transaction effects persist
- The equivalent of contract law

Quote: Jim Gray

"How about interactions among objects in distributed object-oriented system?

Traditionally, these interactions have had vague *maybe-once* semantics -- where a request may be processed zero or one times.

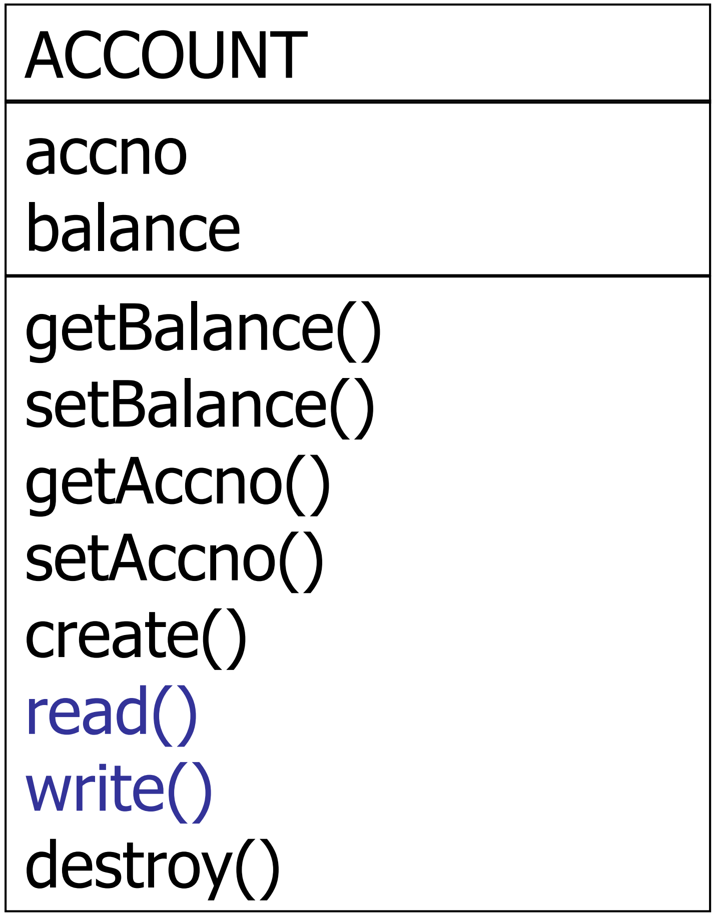
The best model of such interactions is that each interaction is a transaction with some *exactly-once* simple (ACID) guarantees, similar to the guarantees found in the contract law."

Foreword to Bernstain & Newcomer:
Principles of Transaction Processing, Morgan Kaufmann 1997

Bad Example: **not** Proper Design

A class:

Bank Account



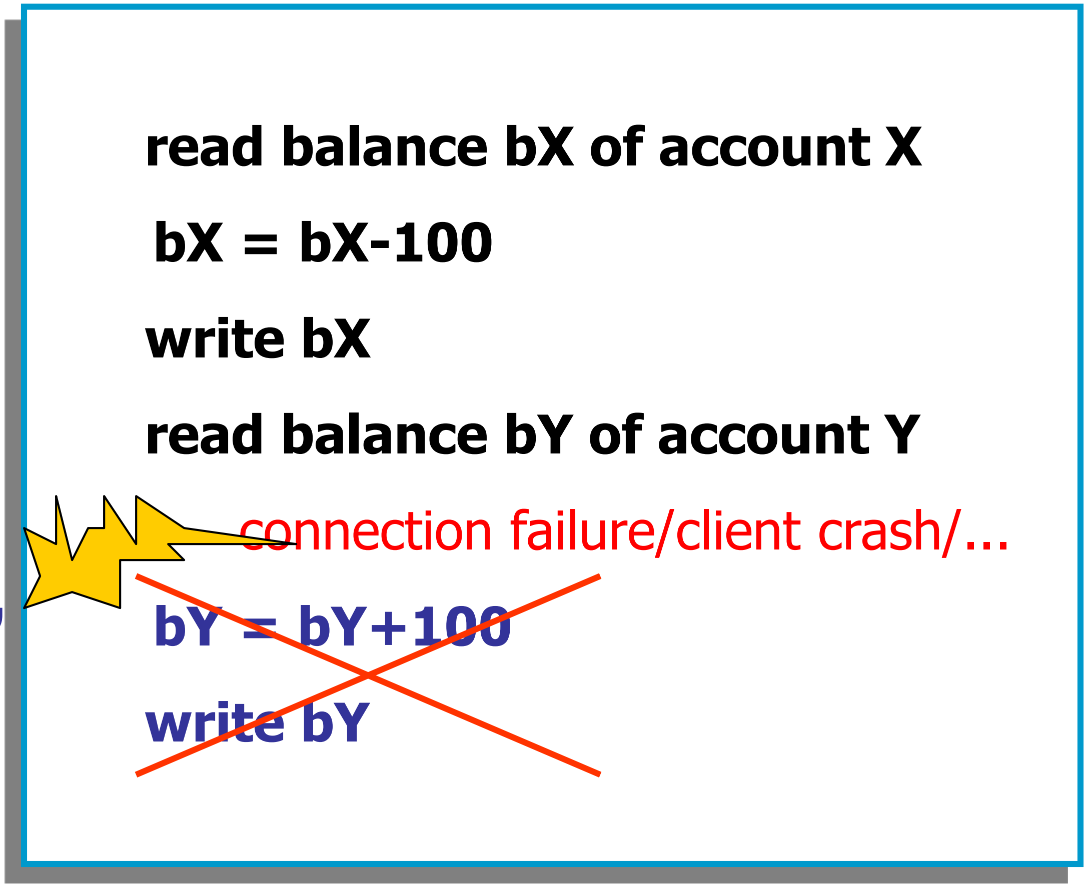
Similar examples in
the textbook
UML Toolkit,
Wiley & Sons, 1998

Failing in Application Design is Risky

Example:

- Transfer 100 € from account X to account Y

- If no transaction, e.g. with **autocommit on**



read balance bX of account X

bX = bX-100

write bX

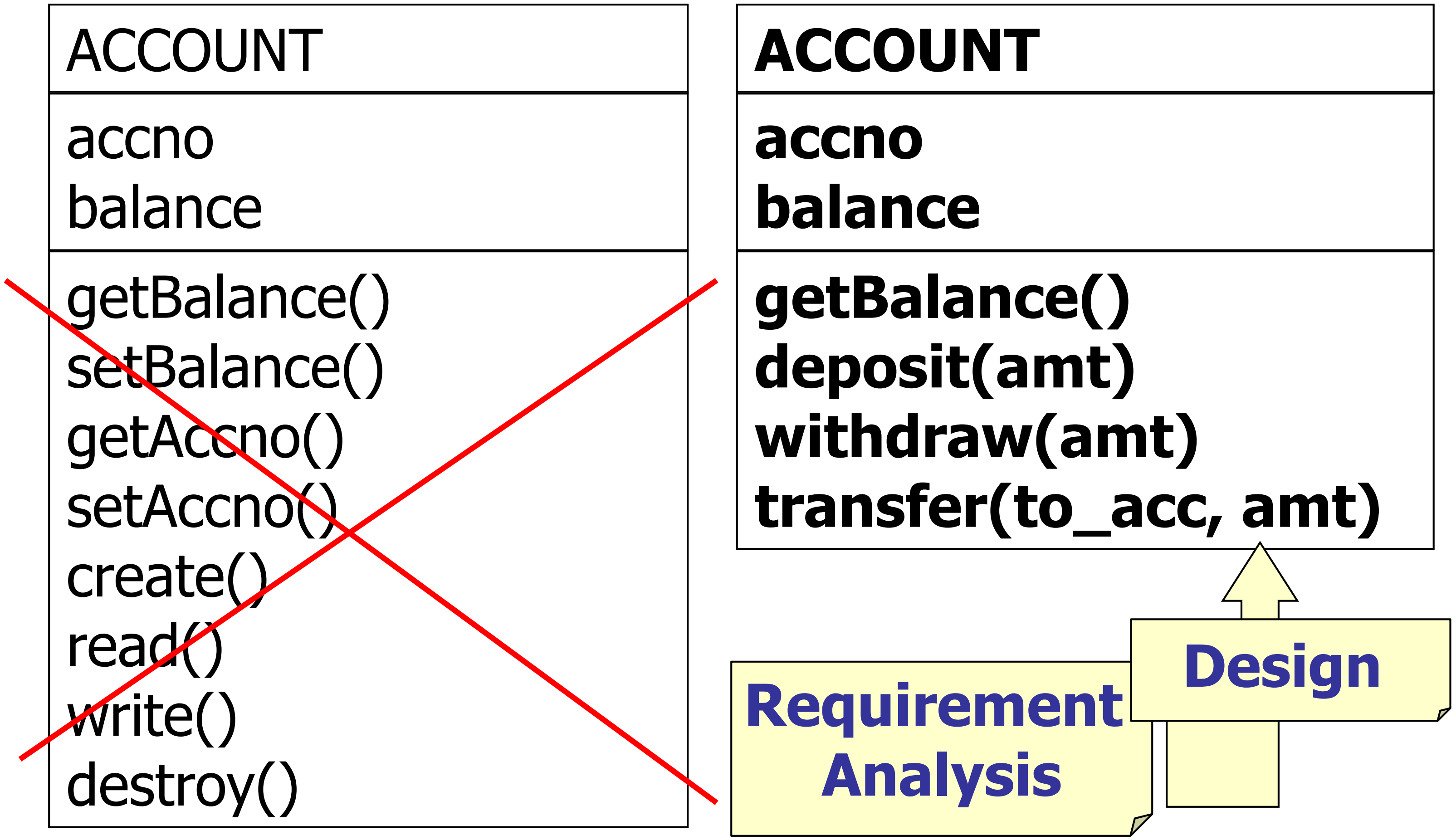
read balance bY of account Y

connection failure/client crash/...

~~**bY = bY+100**~~

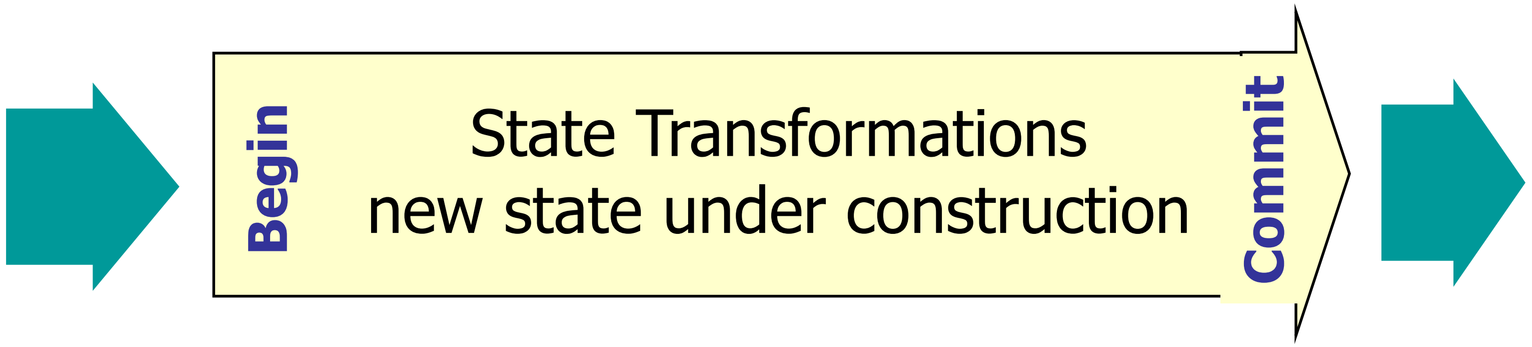
~~**write bY**~~

Example: Proper design



The Issue of Consistency

- Begin-Commit brackets, a set of operations.
- You can violate consistency inside brackets
  - debit but not credit (creates money)
  - create new item before delete old item in a copy
- Begin and Commit are the **points of consistency**



Transactions

Transactions represent basic unit of

- database manipulation.
- database recovery.

The ACID properties

- **A**tomicity: All actions in the transaction happen, or none happen.
- **C**onsistency: If each transaction is consistent, and the DB starts consistent, it ends up consistent.
- **I**solation: Partial effects of incomplete transactions should not be visible to other transactions.
- **D**urability: Effects of a committed transaction are permanent and must not be lost because of later failure.