

SOEN 387 Web-based Enterprise Application Design

Stuart Thiel

Concordia University
Department of Computer & Software Engineering

Fall, 2015

Concurrency

Fowler's
Concurrency
Problems

ACID
Transactions

The Last Mile
On Concurrency
Fixing

Outline

Concurrency

Fowler's Concurrency Problems

ACID Transactions

The Last Mile On Concurrency Fixing

Concurrency

Fowler's
Concurrency
Problems

ACID
Transactions

The Last Mile
On Concurrency
Fixing

Concurrency

- ▶ When things happen at the same time
- ▶ For WEAs, when users access our application at the same time

Concurrency Pitfalls

- ▶ As programmers, we know about deadlock
- ▶ The basis of the problem is really shared data

What is Sharing?

How about separate flows of control accessing same data?

Concurrency

Fowler's
Concurrency
Problems

ACID
Transactions

The Last Mile
On Concurrency
Fixing

How is data shared?

- ▶ static fields?
- ▶ passing messages across threads?
- ▶ client-server?
- ▶ database?

Context sharing problems

- ▶ Who shares application context?
- ▶ Who shares session context?
- ▶ Who shares request context?

Context sharing problems

- ▶ Who shares application context?
- ▶ Who shares session context?
- ▶ Who shares request context?
 - ▶ ... technically the application does too, we usually ignore this

Context Guidelines

- ▶ Don't share data
- ▶ Don't share mutable data
- ▶ Request context isn't sharing, it's just within a thread

How Does the User See Concurrency Problems?

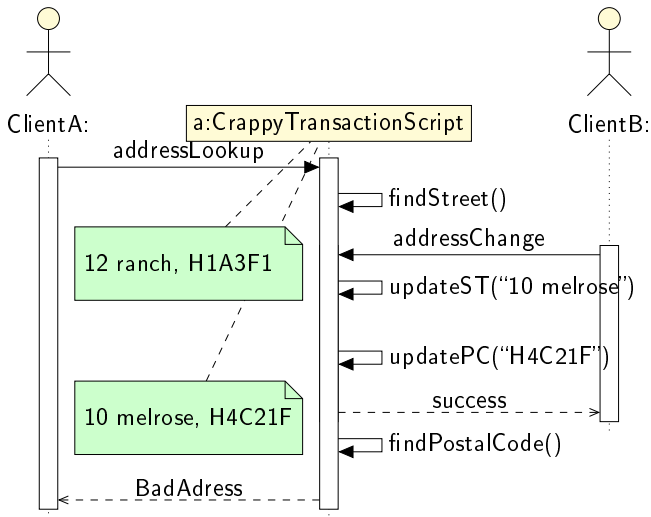
- ▶ Application stops responding?
- ▶ Fowler suggests a couple. . .

Inconsistent Reads

- ▶ Read two correct things in same response
- ▶ They were not correct at the same time

Inconsistent Read Example

What address does Client A get?



Check out [WebSequenceDiagrams.com](https://www.websequencediagrams.com)

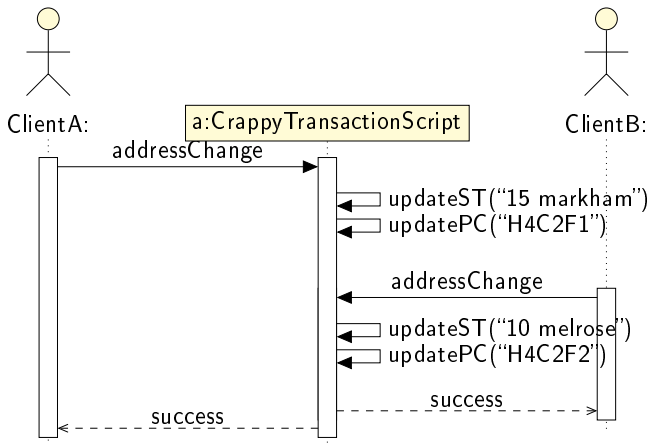
- ▶ <https://www.websequencediagrams.com/>
- ▶ This site is your friend
- ▶ Become proficient and lead happier, more productive lives as Software Engineers
- ▶ ...or whatever for all you Comp Sci students...

Lost Update

- ▶ Two people issue changes at effectively the same time
- ▶ The second change processed overwrites the first
- ▶ The first person does not know that their update has been overwritten
- ▶ The second person does not know that they have overwritten someone else's update

Lost Update Example

What address is saved?



Lost Updates

- ▶ Effectively the same time?
- ▶ What could be shown?
- ▶ How could this effect people?

How Databases Help

- ▶ You **can** share memory
- ▶ You just need to be careful. . .
- ▶ Or you need someone else to have been careful
- ▶ Enter the database and ACID transactions

ACID

- ▶ Atomicity
- ▶ Consistency
- ▶ Isolation
- ▶ Durability

Atomicity

- ▶ Either all actions within a transaction complete, or none of them do

Consistency

- ▶ At the end of a transaction, all integrity constraints for the database hold
- ▶ This does not have to be true during the transaction

Isolation

- ▶ Anything happening within a transaction is invisible outside of the transaction
- ▶ This is often applied in varying degrees
- ▶ What could that mean?

Durability

- ▶ All changes that occur within a completed transaction will “survive permanently”

Transactions

- ▶ How do you start transactions?
- ▶ How do you end them?
- ▶ How can they end?
 - ▶ commit
 - ▶ rollback
 - ▶ others?

Starting Transactions

- ▶ `setAutoCommit()`
- ▶ `commit()`
- ▶ `rollback()`

When to Start a Transaction

- ▶ We've only talked about three patterns:
- ▶ Transaction Script?
- ▶ Template View?
- ▶ RDG?

When to End a Transaction

- ▶ We've only talked about three patterns:
- ▶ Transaction Script?
- ▶ Template View?
- ▶ RDG?

Does ACIDity Fix Everything

- ▶ Does it fix Inconsistent Reads?
- ▶ ... not exactly
- ▶ Does it fix Lost Updates?
- ▶ ... not alone

Inconsistent Reads and Transactions

- ▶ It prevents you reading data made stale by someone else
- ▶ It does not prevent you reading data made stale by your current thread

Lost Updates and Transactions

- ▶ Alone it does not help anything at all
- ▶ ...but we have a reliable and easy solution
- ▶ ...the focus of the next lecture