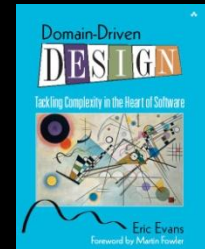


# What I've Learned About DDD Since the Book

August 2003- March 2009

Eric Evans

Domain Language, Inc.



I am a Hands-on Modeller.

# What I've Learned About What is Essential

- Creative collaboration of domain experts and software experts.
- Exploration and experimentation
- Emerging models shaping and reshaping the ubiquitous language.
- Explicit context boundaries
- Focus on the core domain.

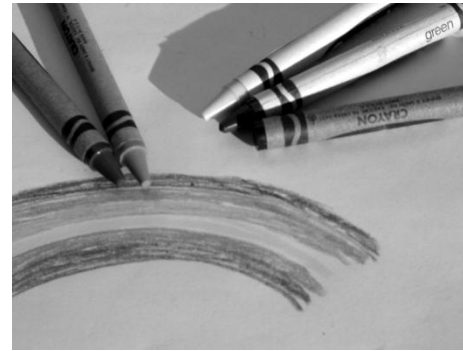
# What I've Learned About Building Blocks



services



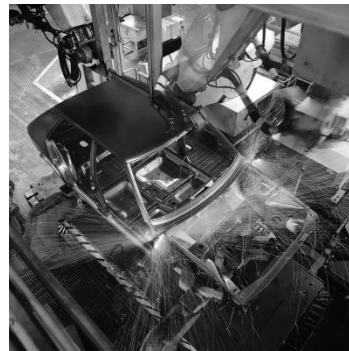
entities



value objects



repositories

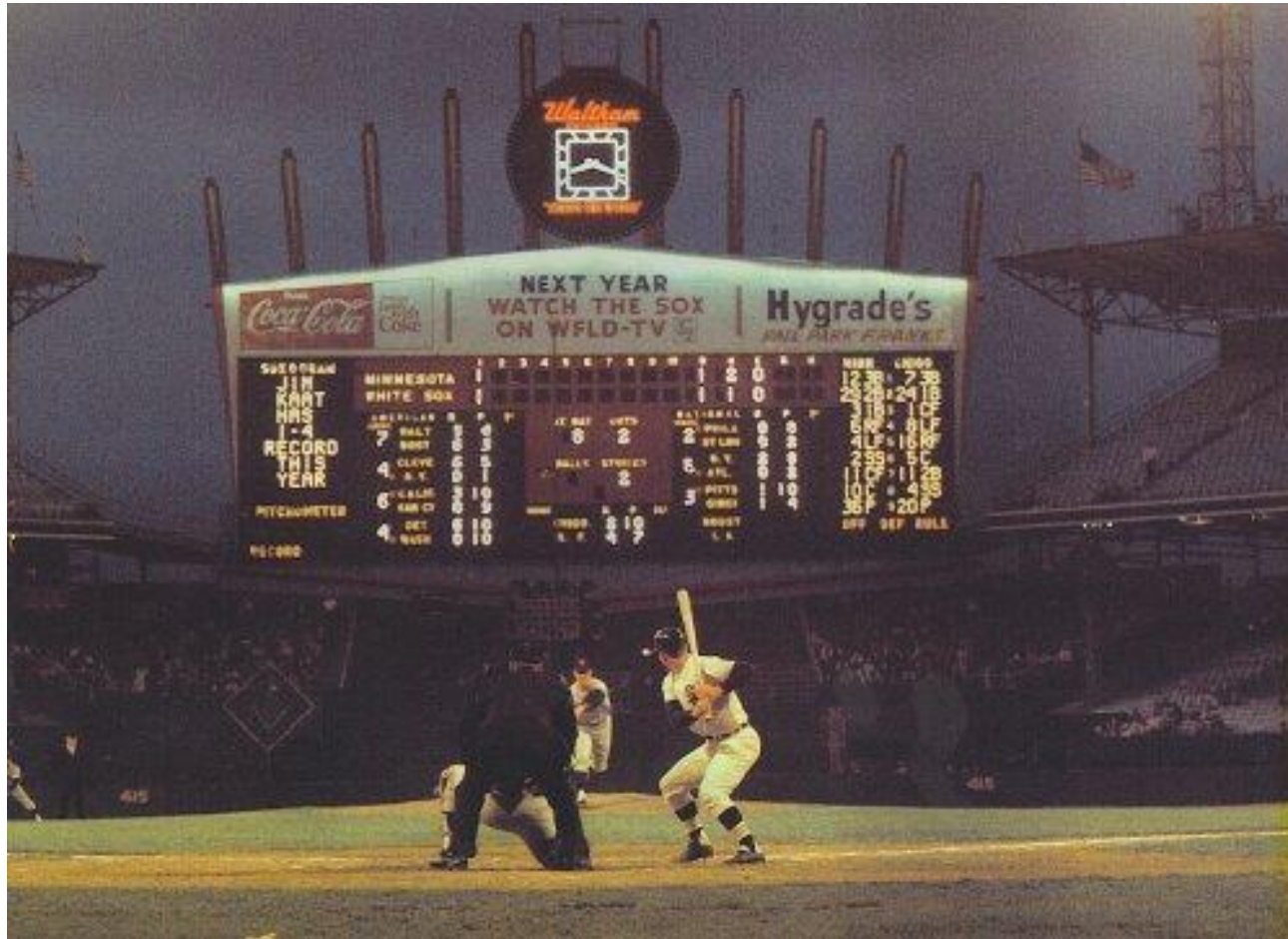


factories

# What I've Learned About Building Blocks

- They are overemphasized.
- But let's add another one anyway!

# Domain Events



Something happened that domain experts care about.

# What do we do with Domain Events

- Clearer, more expressive models
- Architectural options
  - Representing the state of entities
  - Decoupling subsystems with event streams
  - Enabling high-performance system (Greg Young style)

# What I've Learned About Aggregates





# What do we do with Aggregates

- Consistency boundaries
  - Transactions
  - Distribution
  - Concurrency
- Conceptual whole
  - Properties
  - Invariants

# What I've Learned About Strategic Design

- Context Mapping
- Distillation of the Core Domain
- Large-Scale Structure

# What I've Learned About Large-Scale Structure

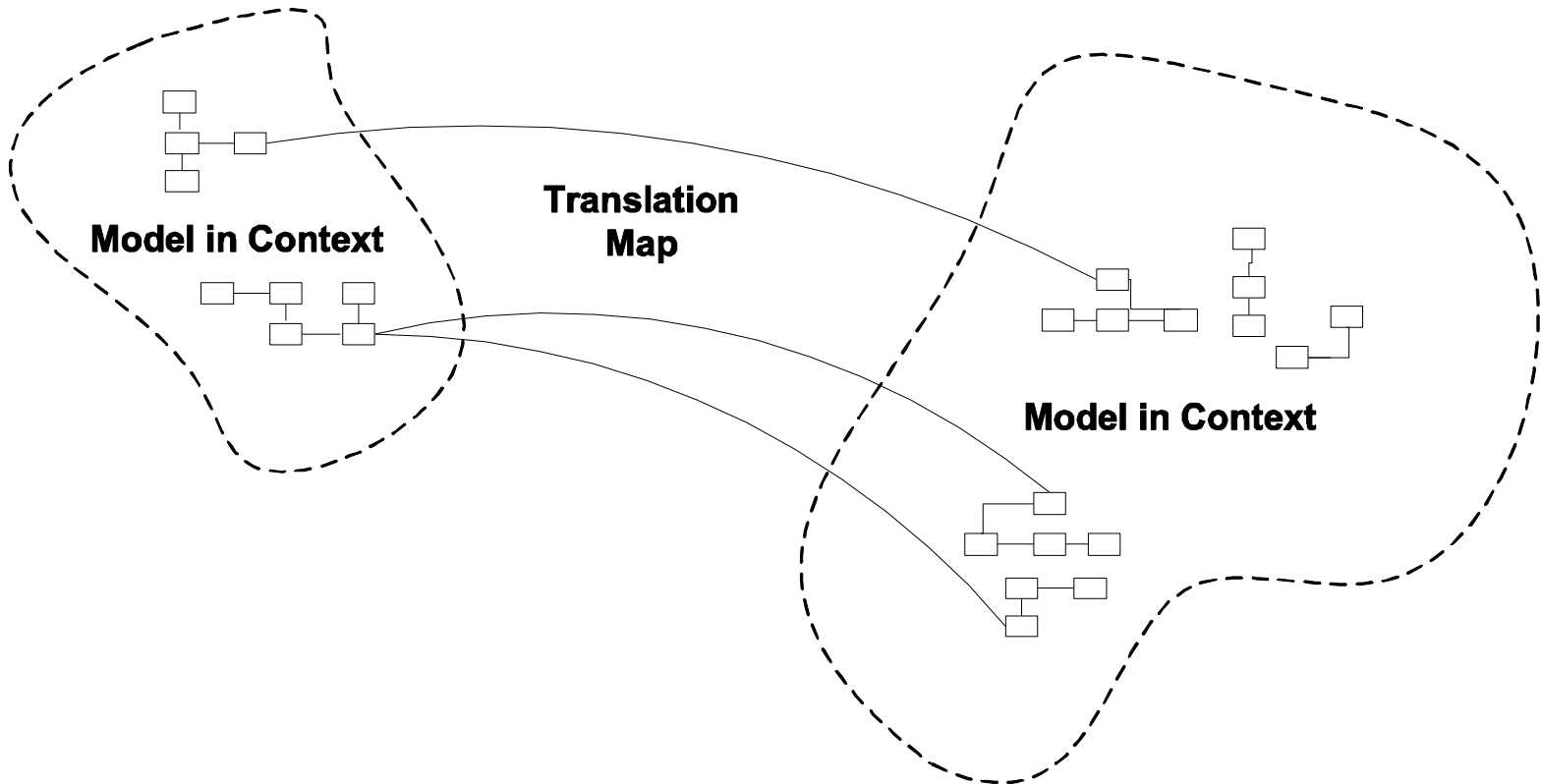


It just doesn't come up very often.

# What I've Learned About Setting the Stage

- Don't spread modelling too thin.
- Focus on the core domain.
- Clean, bounded context.
  
- Iterative process.
- Access to domain experts.

# What I've Learned About Context Mapping



There are always multiple models.

**context** The setting in which a word or statement appears that determines its meaning.

**bounded context** A description of the conditions under which a particular model applies.

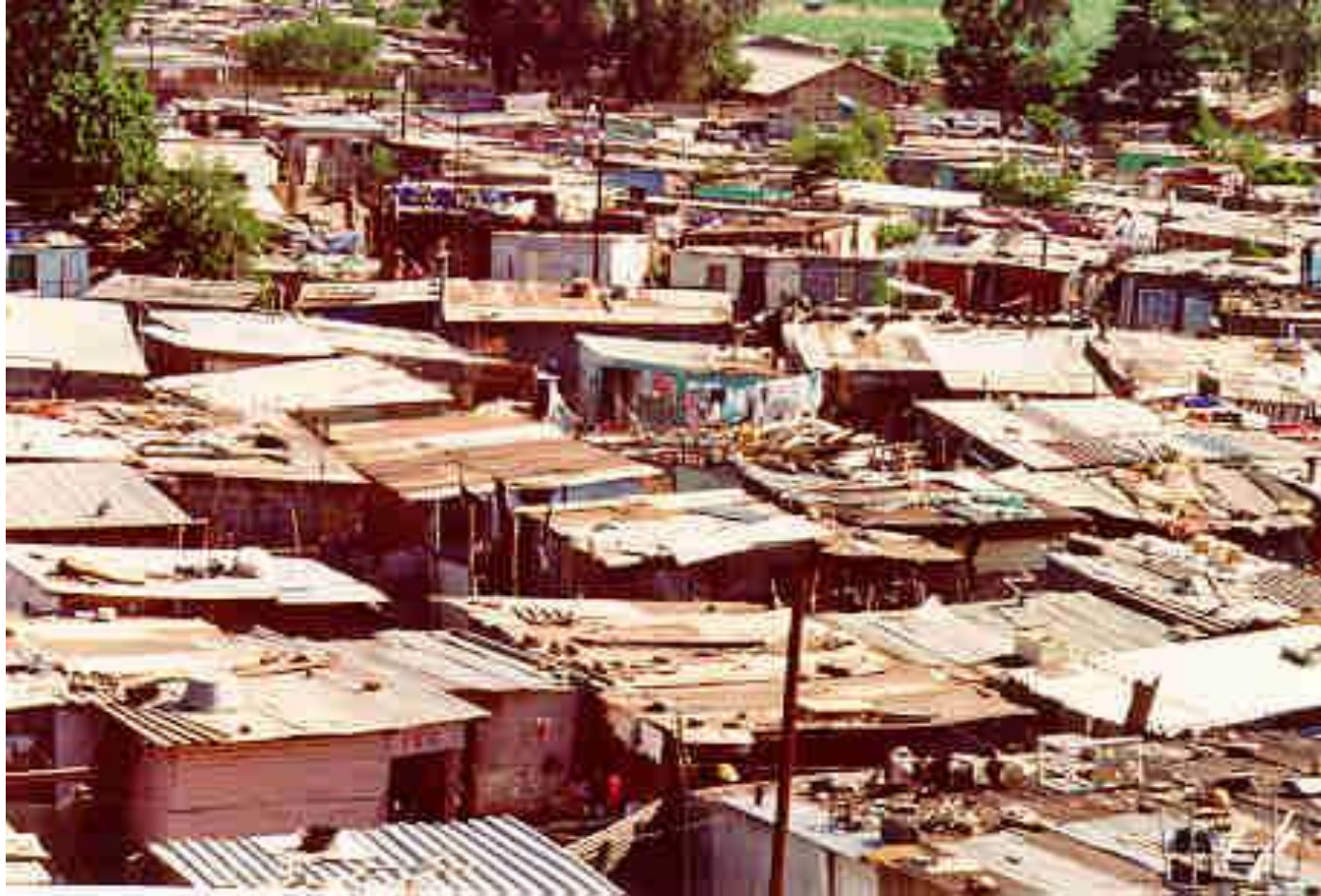
# Partners



- Mutually Dependent
- Cooperative



# Big Ball of Mud



<http://www.laputan.org/mud/>

# Context Mapping

## Step-by-Step

1. What models (or BBoM) do we know of?  
(Draw blob for each and name it.)
2. Where does each apply? (Define boundary in words.)
3. Where is information exchanged? (Draw connection.)
4. What is the relationship?  
(Upstream/downstream? Partner? Etc.)

# Strategy

- Draw a Context Map.
- Work with business leadership to define Core Domain.
- Design a platform that supports work in the Core Domain.
- Work with management to give freedom to the Core Domain Platform Context.
- Develop and model in the Core Domain.

# What I've Learned About DDD and SOA

1. The service interface must be defined in some context.
2. Internals also, but often not the same one.
3. The service interface may define a context boundary.

Precision designs are fragile

Precision designs are fragile

Sophisticated design techniques  
are wasted in a ball of mud.



Anticorruption Layer

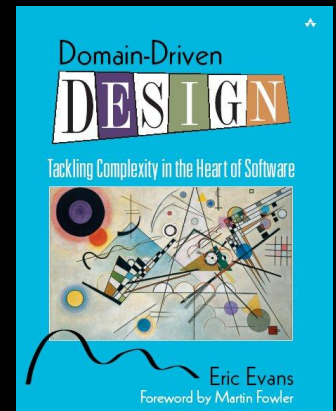
Not all of a large system will be  
well designed.



# DDD Resources

[www.domaindrivendesign.org](http://www.domaindrivendesign.org)

*Domain-Driven Design*  
by Eric Evans



[www.domainlanguage.com](http://www.domainlanguage.com)