ESSENTIAL DOMAIN-DRIVEN DESIGN (DDD)

Participant Workbook

virtualgenius
leading by design

Opener

Introduce yourself to someone from a different table group and share with each other what you already know about software design.

Return to your table group and share what you learned in your interviews. What common themes do you notice?

Domain Modeling – Case Study

The Dishwasher

Four years ago, on Oct 10, David and Claudia purchased a brand-new Whirlpool dishwasher. It was an expensive dishwasher, but a family with six children has a lot of dishes. They wanted a reliable, high-capacity dishwasher, and so they spent quite a bit of money on one that would do the job. The dishwasher had a retail price of $1,399 USD, but they were able to get it on sale for $1099 USD. It came with a 3-year manufacturer's warranty.

Our company, the *Elan Extended Warranty Company*, sends out a mailing just prior to the expiration of each customer's manufacturer's warranty. We keep those contact records in our Leads database, and we send each contact a letter explaining that for a small amount of money we can extend the warranty coverage over that product for another 3 years. David and Claudia thought, "Wow, we did spend a lot of money on this, and if we had to replace this right now, that's a big outlay for us. So let's pay a little extra for this warranty and have the peace of mind it will buy us." The pain of that initial purchase 3 years ago is in the background now, so that additional outlay for the extended warranty didn't seem as much to them.

The mailing went out on July 1. And on Oct 11, after responding to that mailing, David and Claudia became the proud owners of one of our company's extended warranties over their dishwasher. They became our customers.

On Nov 5, David called our call center and said the dishwasher is not working. What everyone fears actually happened, the moment the manufacturers warranty expired the product stopped working. The customer service representative (CSR) at the call center, John, opened a claim after he looked up David and Claudia's contract. He found their customer information, opened a new claim, and said, "What does the problem seem to be?" David said, "The problem is, as far as I can tell, that the soap dispenser is not opening during the wash cycle. No soap, dirty dishes. 6 children. Piles of dirty dishes."

John said, "No problem. I see this is a covered product under the extended warranty. The Nov 5 failure date of the product is after the Oct 11 effective date of the contract, and the Oct 10 expiration date is still several years away." So John creates a purchase order (PO), which is sent to a 3rd party-system managed by a servicer management company.

On Nov 12, Rick the dishwasher repairman comes out. David says, "The soap dispenser doesn't seem to be working." So Rick replaces the soap dispenser, runs the dishwasher and it's working.

Three days later, on Nov 15, David calls the call center again and talks with Jane the CSR. The dishwasher is still not working. David explains that the day after Rick left, the same problem occurred again, and has continued to happen. Jane looks up the contract and sees that the claim for David's contract has been closed because Rick had said it was fixed. So Jane reopens the claim, and creates a new PO.

Rick returns to David and Claudia's house on Dec 1st, and replaces the soap dispenser a second time. In the process of replacing the soap dispenser for a second time, he breaks one of the components inside the dishwasher and has to replace that as well. The details for the components and their costs went on the second repair PO.

On Jan 15 Claudia has to make the call because David can't stand going through the process again. She says to Claire the CSR, "It's still not working. Help! Please don't send the same person. Rick was nice, but all he keeps doing is replacing the soap dispenser and the second time he replaced it he broke something else, so can you send someone different?" So this time we made sure a repairer was dispatched who was an expert in this particular brand and model of dishwasher. This repairman, Vince, was hard-core: he replaced the motherboard. He figured, based on

running a diagnostic, that it was a software problem. He replaced the motherboard, and a couple of other products inside there, just in case. Before he left he ran the dishwasher and it worked just fine. That was on Jan 30. This has been going now since Nov 5, and this was David and Claudia's third repair.

On Feb 5th, David calls the call center again and speaks to Debbie the CSR and says, "I'm done." I don't want any more repairs on this product. Debbie says, "Wait. There's only been one claim, which is closed, but I see three Repair Purchase Orders on it. What is going on here? I see we're spending a lot of money on repairing this covered product, and it is approaching the point where more has been spent on the repairs than the product is actually worth - our contract is reaching our limit of liability."

Debbie says, "I'm going to reimburse you for the retail value of the dishwasher. I'll give you store credit for the market replacement value of the dishwasher so you can go purchase a new one from the store."

Debbie closed the claim, and terminated the contract, since it is fulfilled at this point (we've met all our legal obligations so there is nothing left to do on the contract). She emailed the Finance department to mail out the store credit.

So David and Claudia took their store credit and on March 10th purchased a new Whirlpool dishwasher with a three-year manufacturer warranty. The saga starts all over again, but at least now they have clean dishes again![1]

Review the story again, underlining anything you think might be important for understanding and modeling the business domain.

Circle any words that might be key concepts for this domain.

---

[1] *This is a true story, though some details such as names and dates have been changed.*

Case Study – Review Questions

1. How many PO's were created, and what kind of information did they contain?

2. What key steps were involved in closing the claim for the final time?

3. What are the top three domain concepts in the story?

4. Our extended warranty company wants to "add some more smarts in the software" to the product replacement process. What details in the story might be most significant for that enhancement?

5. How does walking through a concrete business example like this help you understand the business domain?

6. How would a concrete example like this help with automated acceptance testing?
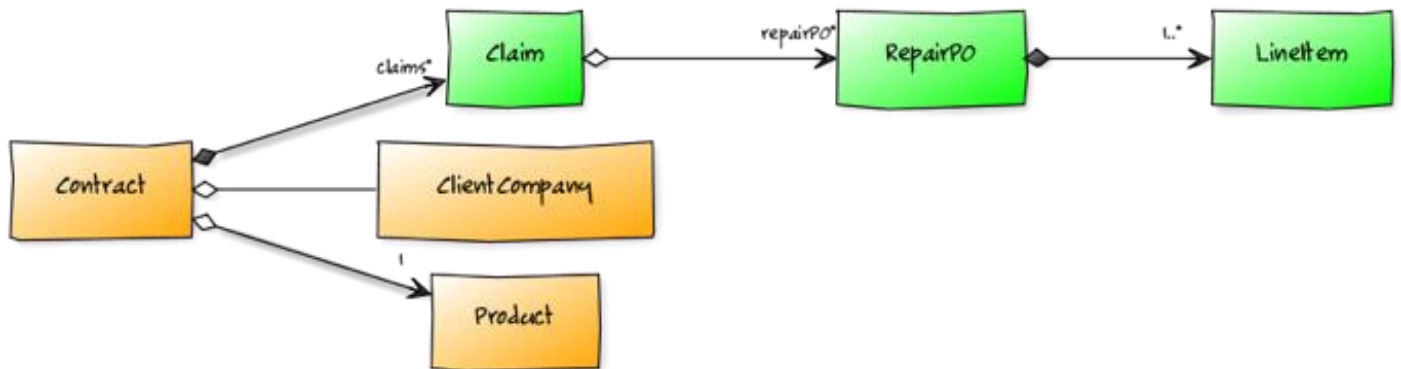
www.virtualgenius.com

Ubiquitous Language

Summarize your understanding of "ubiquitous language" in DDD.

What are the benefits of growing a ubiquitous language for your project?

Where are places in your development work where you could benefit from growing a robust ubiquitous language?
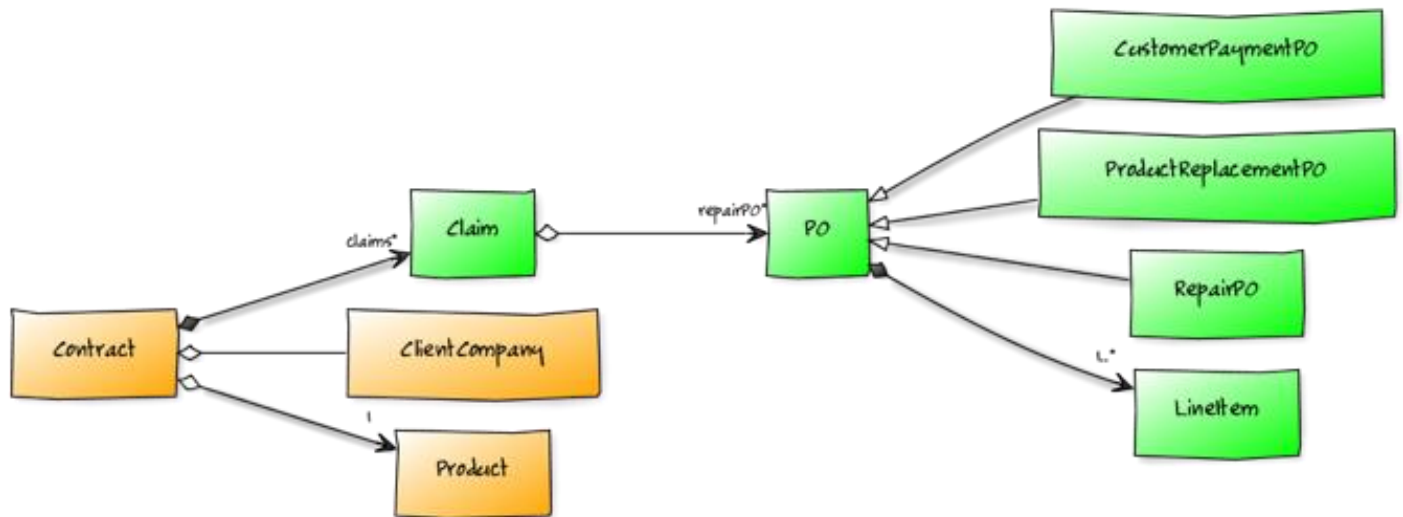
Case Study – Domain Model

Version 1.0

Draw an instance diagram that reflects the state of the system at the end of the case study.

www.virtualgenius.com

Proposed Model

Compare and contrast the current (v1.0) domain model with the proposed model. How well does the proposed model reflect the ubiquitous language you identified in the case study?
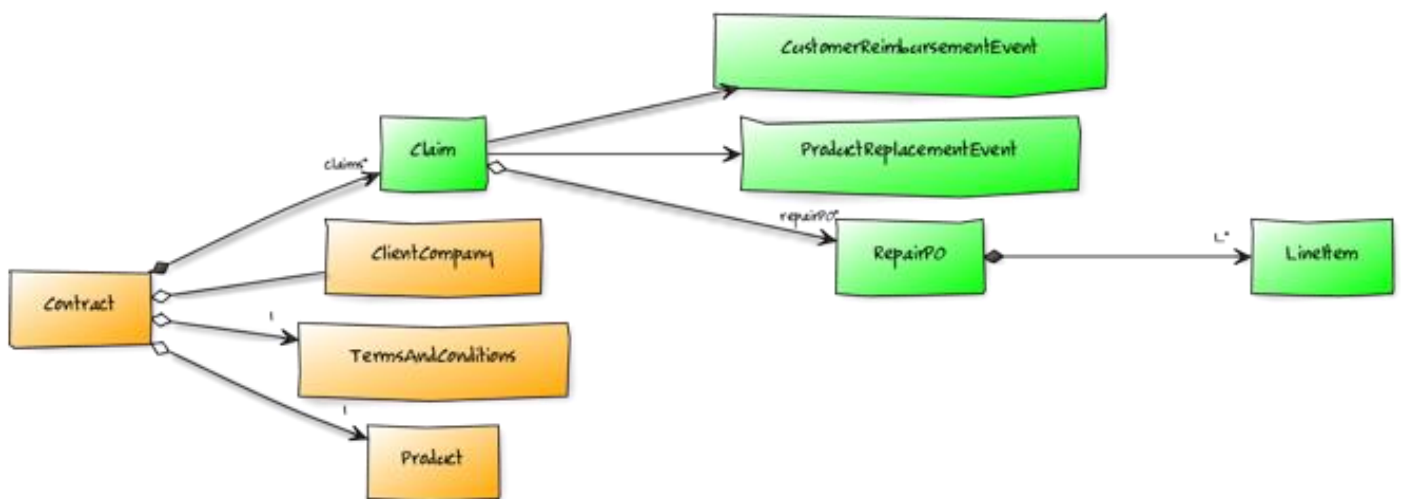
Model Exploration

Draw the model exploration whirlpool below

Tactical Design

Domain Events

Write down at least three key things to remember for domain events

Case Study – Proposed Model 2

How do the domain events in the model above make it a more useful and robust model?

What improvements do you notice in the ubiquitous language in the proposed model above?

Value Objects and Entities

Capture the concept-maps about value objects and entities below

Why should we be biased towards modeling concepts as value objects?

What did the instructor mean by making value objects centers of gravity for behavior?

Aggregates

Summarize the aggregate design rules of thumb as a concept map below

What are the consistency rules for aggregates?

Repositories

What is the relationship between aggregates and repositories?

Basic DDD Tactical Modeling Patterns

Entities

- Have identity (must be able to identify one from another) and lifecycle (change over time)
- Equality by unique identifier (e.g. contract # for extended warranty contract)
- Mutable (can be changed), therefore "unsafe"
- Keep them focused on identity and lifecycle only
- Avoid them getting "bloated" by using value objects - entities delegate the bulk of their behavioral work to value objects

Value Objects

- No identity - equality by properties (e.g. two dates with same day, month & year are the same date)
- Give rich, expressive names to key concepts in our model
- Increase the conceptual cohesiveness of objects, readability of our code and the overall suppleness of our design
- Making value objects immutable reduces incidence of bugs, improves thread safety and encourages a functional composition style
- Encapsulation (information hiding) reduces cognitive burden and increases communication
- Reduces "primitive obsession"
- Removes burden of maintaining state from entities, focusing them on identity and lifecycle

Domain Events

- Something happened that the domain expert cares about
- Enable us to make explicit events that we want to track or be notified about, so they are very useful for modeling time-based things
- Can trigger other domain events
- Typically named in the past tense (e.g. `OrderProcessedEvent`)
- Immutable, since they are a record of something that happened in the past

Aggregates

- Collection of entities and value objects treated as a conceptual whole
- Consistency rules:
  - Logically consistent *within* an aggregate (invariants apply at transaction commit)
  - Eventually consistent *across* aggregates (updates propagate asynchronously)

Repositories

- Domain service for saving and retrieving whole aggregates from persistence mechanism

www.virtualgenius.com

Strategic Design

Domain Distillation

Draw the Purpose Alignment Model and how it maps to DDD strategic domain distillation below.

Context Mapping

Draw summary diagrams of the various types of context relationships below

Event Sourcing

Summarize your key insights on event sourcing below

Hexagonal Architecture

Draw a diagram describing hexagonal architecture

What does the hexagonal architecture approach emphasize compare to a layered architecture approach?

What might be possible challenges with implementing a hexagonal architecture?

How does hexagonal architecture positiely impact the testability of a system? What are the other key benefits?

CQRS

Draw a diagram describing command-query responsibility separation.

What are the key benefits of CQRS?

How is CQRS different from event sourcing?

Learning Log – Day 1

Here is your place to capture your "aha's" from today. Include such things as:

- things you learned,
- new insights you want to remember,
- cool things you heard other people say, and
- things you want to experiment with based on what you experienced.

Learning Log – Day 2

Here is your place to capture your "aha's" from today. Include such things as:

- things you learned,
- new insights you want to remember,
- cool things you heard other people say, and
- things you want to experiment with based on what you experienced.

Learning Log – Day 3

Here is your place to capture your "aha's" from today. Include such things as:

- things you learned,
- new insights you want to remember,
- cool things you heard other people say, and
- things you want to experiment with based on what you experienced.

About Virtual Genius LLC

Virtual Genius LLC helps software teams succeed, with customer-focused collaborative design at the center of everything the team does. This approach emphasizes principles and practices from design thinking, domain-driven design (DDD), user-experience design (UXD), and behaviour-driven development (BDD).

We provide world-class training and coaching (both onsite and remote) in BDD and DDD. Contact us if you want to take your team's skills to the next level.

Contacting Paul

Blog: www.thepaulrayner.com
Twitter: @thepaulrayner
Email: paul@virtualgenius.com
Phone: 1-303-374-4370 (USA)

www.virtualgenius.com