Domain-Driven Design
Modeling An E-Commerce App

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

About Me

I have more than 24 years of experience in software development, being involved in various projects covering desktop, client/server, web, cloud, and mobile applications using mainly Microsoft tools & technologies, but also Apple and open-source.

I'm a Certified Scrum Master and a certified APMG Agile Project Manager. Since 2003, I'm holding a Ph.D. degree in Industrial Engineering.

You can find more about me on LinkedIn:
https://www.linkedin.com/in/eduardghergu/

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Domain-Driven Design: A Deep Dive

1. A summary of the gathered knowledge

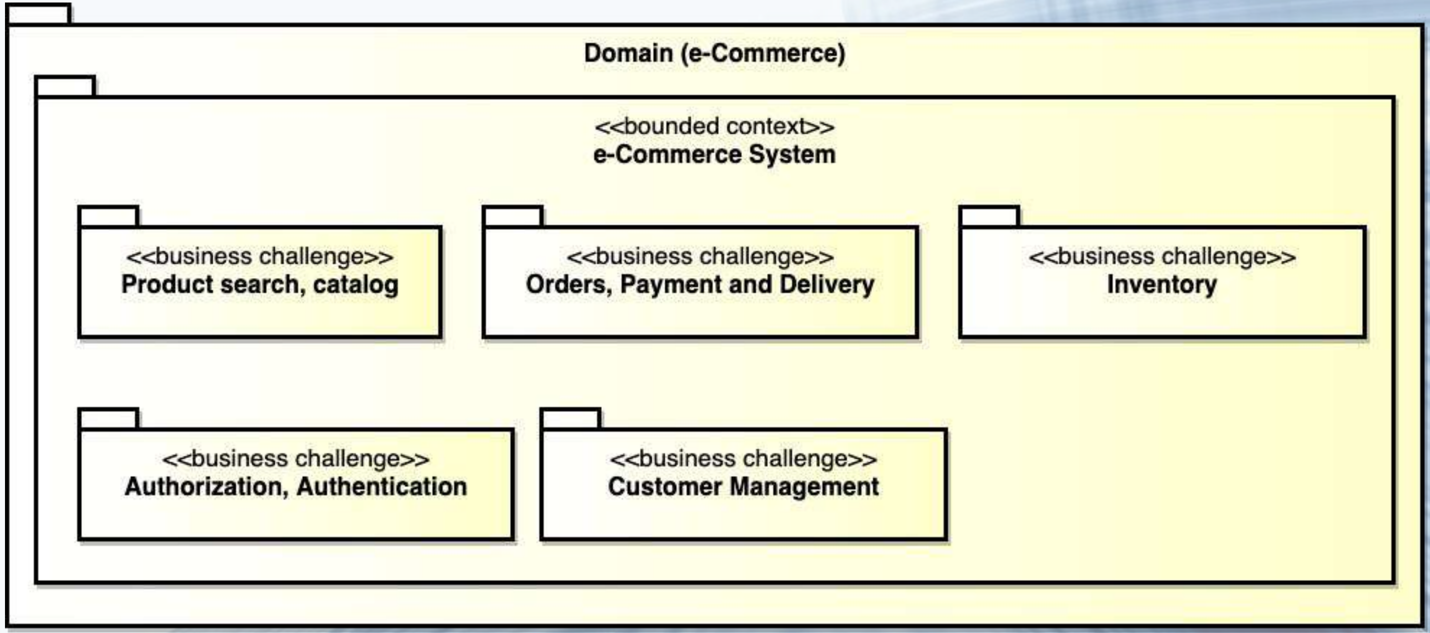
2. Work on a sample application

3. Conclusions. Q&A

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

The gathered knowledge

Let's review a sample e-commerce domain, before applying DDD [12]...



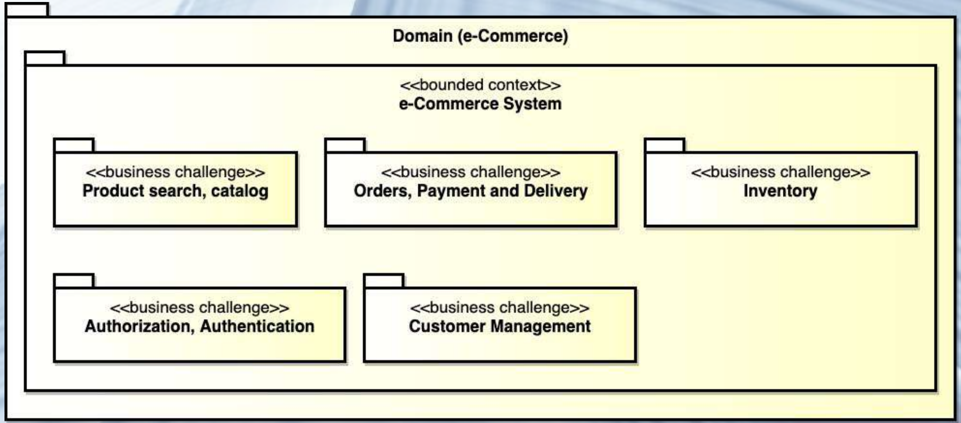
Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

The gathered knowledge - cont.

Let's start applying some DDD to this problem…

1. Identify the possible Subdomains inside a Domain (the "problem" to solve). A Domain has its own strategic challenges which can be seen as Subdomains.

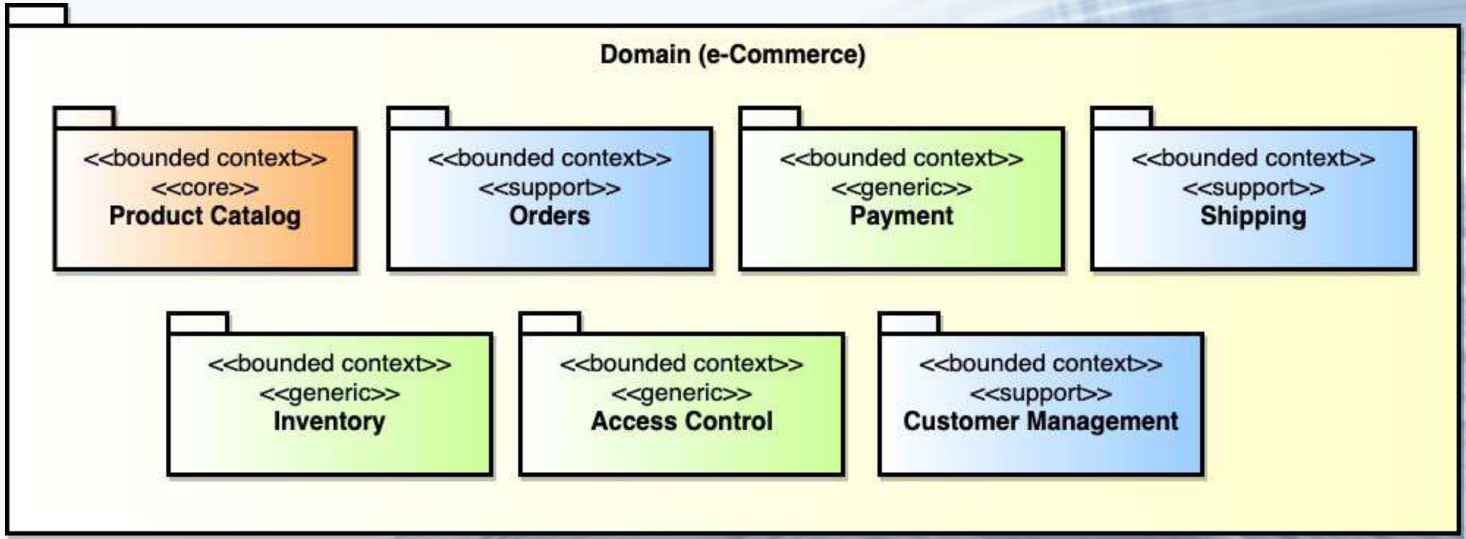
2. Split the Domain in Subdomains. It is a good practice to set a Bounded Context for each Subdomain.



Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

The gathered knowledge - cont.

The result [12]:


Domain (e-Commerce)
<<bounded context>> <<core>> Product Catalog
<<bounded context>> <<support>> Orders
<<bounded context>> <<generic>> Payment
<<bounded context>> <<support>> Shipping
<<bounded context>> <<generic>> Inventory
<<bounded context>> <<generic>> Access Control
<<bounded context>> <<support>> Customer Management

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

The gathered knowledge - cont.

The identified Bounded Contexts are classified as Core, Support and Generic:

- **Usually, there is only one Core, that represents the main part of the Domain - the *Product Catalog*.**
- **Support Bounded Contexts are auxiliary ones. They will support the Core, however, they will have their own model.**
- **Generic Bounded Contexts are similar to support ones, but they have a strong particularity: they are pretty generic so that they could be used not only in the Domain they were created, but they could be used by others Domains, too.**

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app

Requirements and Modelling:

- Identify User Stories
- Identify the Nouns in the user stories -> Entities
- Identify the Verbs in the user stories -> Behaviour
- Identify the objects interaction (UML diagram)
- Identify the objects responsibilities (UML diagram)
- Create the UML class diagram (only the main relationships)

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling

User Stories:

- As a customer, I want to be able to add the products that I want to purchase to the shopping cart, so that I can checkout quickly later on
- As a customer, I want to see the cost of each item in the shopping cart, so that I can re-check the items' price
- As a customer, I want to see the total cost for all of the items that are in my cart
- As a customer, I want to see the total cost for all of the items in the shopping cart with total tax

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

User Stories - cont.:

- As a customer, I want to be able to specify the address of where all of the products are going to be delivered

- As a customer, I want to be able to add a note to the delivery, so that I can provide special instructions to the courier

- As a customer, I want to be able to specify my credit card information during checkout, so that I can pay for the items

- As a customer, I want the system to tell me how many items are in stock, so that I know how many items I can purchase

Sample app - Req. and Modelling - cont.

User Stories - cont.:

- **As a customer, I want shopping cart to check that items are still available for purchase during checkout, so that I can still purchase items that are in the cart**

- **As a customer, I want to receive order confirmation email with order number, so that I have the proof of purchase**

- **As a customer, I want to specify the invoice address for an order, so that I can receive the invoice for that order**

And the list can continue...

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

Nouns:

- Customer
- Item
- Order
- Shopping Cart
- Address
- Invoice
- Delivery
- Tax
- Credit Card Information

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

Verbs:

- Put products in to the shopping cart
- See total cost for all of the items
- See the cost for each item
- See total tax for my country
- Specify delivery address
- Specify delivery note for delivery address
- Specify invoice address

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

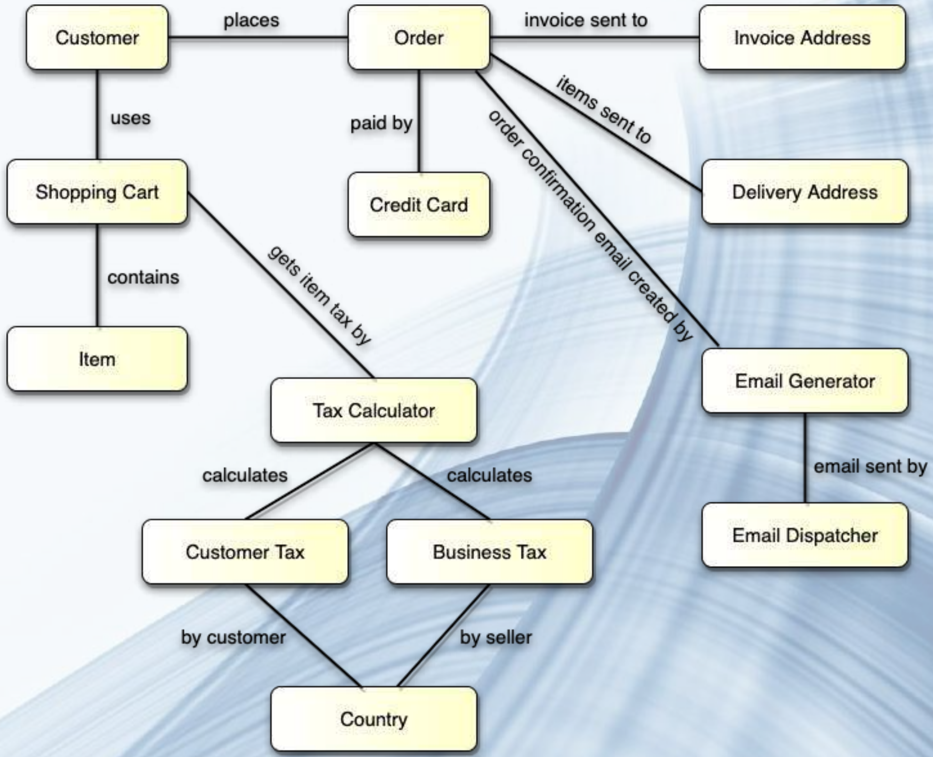
Verbs - cont.:

- Receive invoice for the order
- Sent invoice
- Specify credit card information
- Pay for the items
- Tell me how many items are in stock
- Check that items are still available during check out
- Receive order confirmation email

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

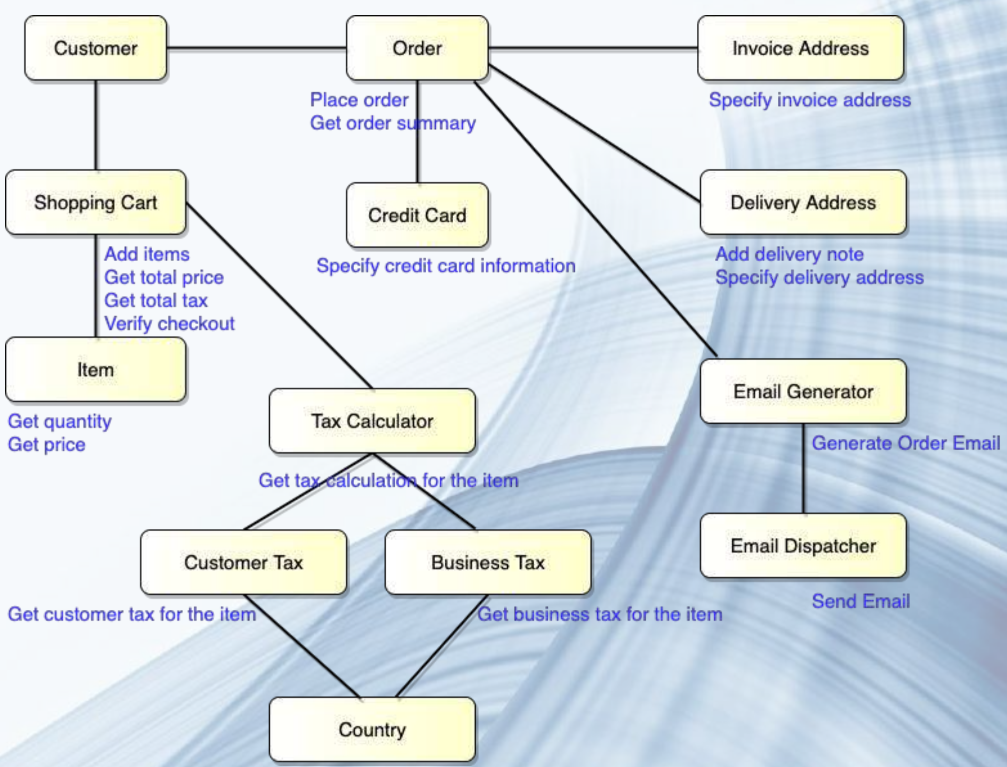
Objects interaction:



Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

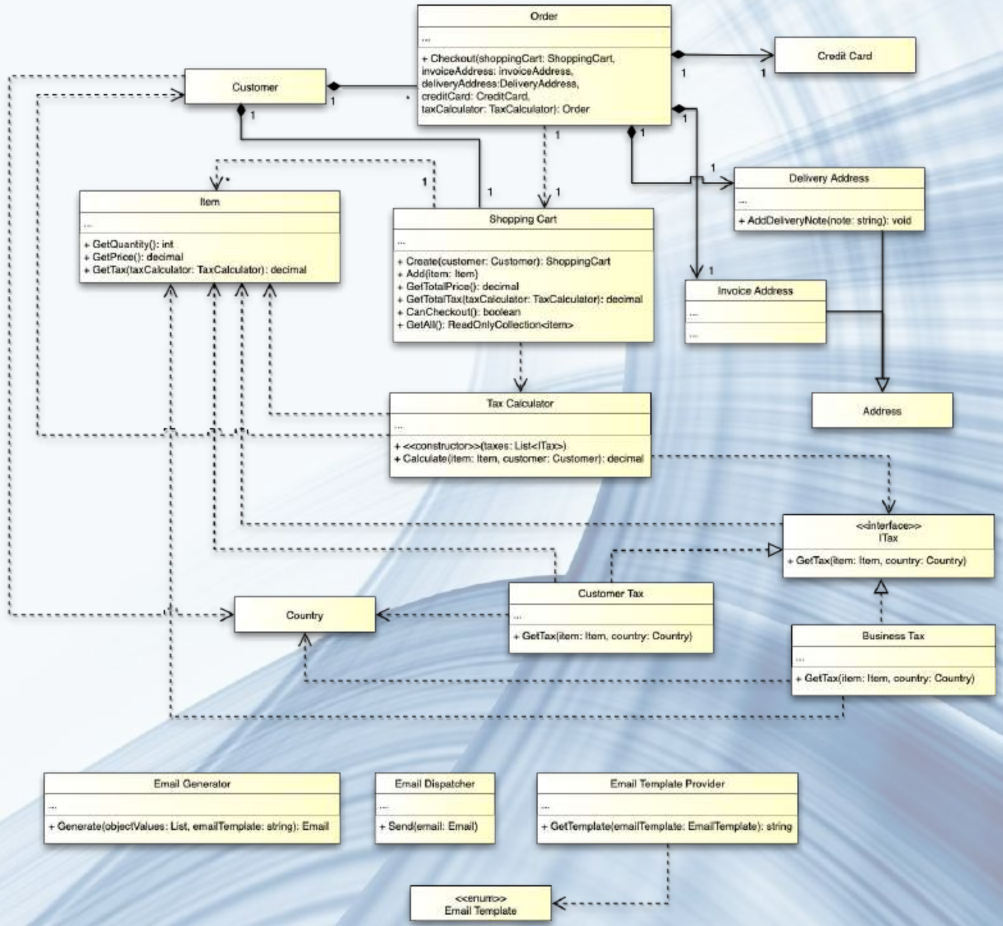
Objects responsibilities:



Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

Class diagram:



Order
...
+ Checkout(shoppingCart: ShoppingCart, invoiceAddress: invoiceAddress, deliveryAddress:DeliveryAddress, creditCard: CreditCard, taxCalculator: TaxCalculator): Order
Credit Card
Customer
Delivery Address
...
+ AddDeliveryNote(note: string): void
Item
...
+ GetQuantity(): int
+ GetPrice(): decimal
+ GetTax(taxCalculator: TaxCalculator): decimal
Shopping Cart
...
+ Create(customer: Customer): ShoppingCart
+ Add(item: Item)
+ GetTotalPrice(): decimal
+ GetTotalTax(taxCalculator: TaxCalculator): decimal
+ CanCheckout(): boolean
+ GetAll(): ReadOnlyCollection<item>
Invoice Address
...
...
Address
Tax Calculator
...
+ <<constructor>>(taxes: List<ITax>)
+ Calculate(item: Item, customer: Customer): decimal
<<interface>>
ITax
+ GetTax(item: Item, country: Country)
Customer Tax
...
+ GetTax(item: Item, country: Country)
Country
Business Tax
...
+ GetTax(item: Item, country: Country)
Email Generator
...
+ Generate(objectValues: List, emailTemplate: string): Email
Email Dispatcher
...
+ Send(email: Email)
Email Template Provider
...
+ GetTemplate(emailTemplate: EmailTemplate): string
<<enum>>
Email Template
1
1
*
1
1
1
1
1
1
1
1
1
1
*

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Req. and Modelling - cont.

Summary:

- **Don't start doing anything until you have requirements**
- **Don't just jump in to the code soon as you have requirements; put together object interaction and responsibilities diagrams first.**
- **When you have identified your objects, interactions and responsibilities use UML class diagrams to put together a draft model**
- **Don't try to model the reality of the world, model the reality of your organisation - different companies will have different approaches. Remember, it is all about your business domain and not the actual "reality".**

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Sample app - Code (draft)

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Q & A

For any inquiries or requests:
eduard.ghergu@professional-programmer.com

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

Thank you!
See you at the next webinar!

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

References

1. https://www.infoq.com/minibooks/domain-driven-design-quickly/
2. https://vladikk.com/2016/04/05/tackling-complexity-ddd/
3. https://en.wikipedia.org/wiki/Domain-driven_design
4. https://www.jamesmichaelhickey.com/clean-architecture/
5. https://blog.knoldus.com/is-shifting-to-domain-driven-design-worth-your-efforts/
6. https://thedomaindrivendesign.io/
7. https://www.culttt.com/2014/11/12/domain-model-domain-driven-design/
8. https://martinfowler.com/bliki/BoundedContext.html
9. https://github.com/zkavtaskin/Domain-Driven-Design-Example

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com

References - cont.

10. https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

11. https://www.freecodecamp.org/news/a-quick-introduction-to-clean-architecture-990c014448d2/

12. http://www.fabriciosuarte.com/2016/02/domain-driven-design-hands-on-example.html

Eduard Ghergu, Eng., PhD.
www.professional-programmer.com