

www.EdwinvanDillen.nl

Sogyo



DDD en DSL: een mooie combinatie!

Session Code: CT.01



Objective of this session

Explore the concept of Domain Specific Language (*specific the Microsoft interpretation and implementation*) and see how DSL can be applied in a Domain Driven Design software development approach.



How to fulfil the objective?

- Bird view of DDD
- Introduction in DSL
- Bring DDD and DSL together
- An example: DSL for persistency
- Conclusions



Bird view of DDD

The shorts introduction of DDD
you have ever seen...



What is domain modeling about?

- “Domain modeling is not a matter of making as ‘realistic’ a model as possible. Even in a domain of tangible real-world things, our model is an artificial creation....”
- “It is more like movie making, loosely representation reality to a particular purpose”
- Create a common language;

Source: “DDD: tackling complexity in the heart of software” by Eric Evans



The context of DDD

- Domain Driven Design by Eric Evans
 - Mainly focused on the design
- Domain Driven Design/Development by Sogyo
 - Beyond design, also address implementation and software architecture;
 - Stronger relation with OO;
 - Behavioural domain;



Introduction in DSL

From the background of DSL to
the Microsoft DSL tools
implementation



Domain Specific Languages

- What does Domain mean here?
 - “any topic of interest” Keith Short
 - Domains are used to get the focus on a specific topic within the whole area of the system
 - Make solutions easier to understand and maintain
 - Easy to integrate with development process
 - Improve agility through rapid iteration
 - Domain examples:
 - Security (rather broad domain)
 - Web service security (more specific domain)
 - Also business domains, such as Health insurance



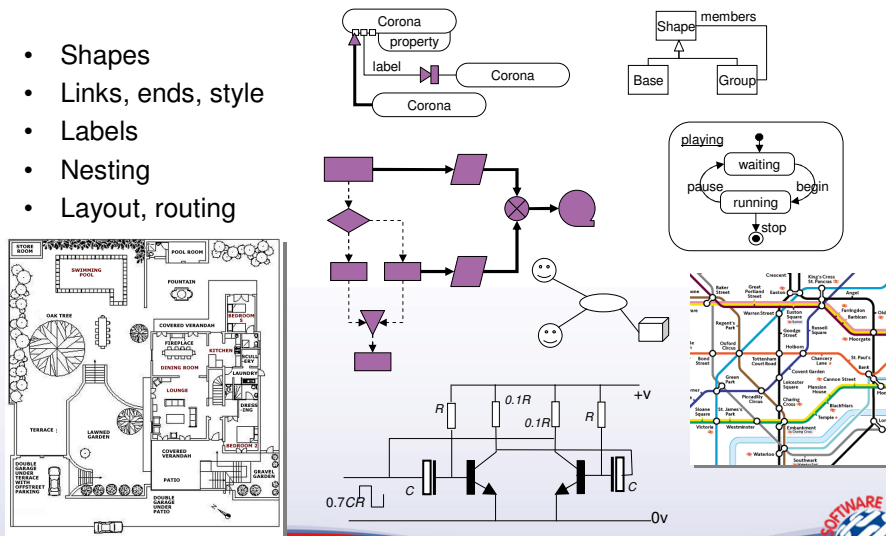
What is a DSL

- For instance, a DSL to build Web Services;
- Normally we need:
 - XML
 - .ASMX (WSDL)
 - C#
 - .config
- A DSL for Web Services makes sure that all these parts are represented visually as one;
- This makes it possible to check-out one specific web services by just clicking on it.



Visual Languages

- Shapes
- Links, ends, style
- Labels
- Nesting
- Layout, routing



Why visual languages

- To help understand or define requirements
- To help understand or define designs
- To generate parts of the solution
- To implement design patterns for specific frameworks and architectures
- To customize applications and application components
- To visualize existing systems



Benefits of Domain Specific Languages

- Communication and comprehension
 - Focus on domain abstractions
 - Readability
 - Clear mental model
 - Communication with customers
- Explicit grammar
 - Constraint validation



Benefits of Domain Specific Language

- Suitable modern editor
 - Prompting and contextual guidance – what is and is not allowed
 - Navigation and searching
 - collapse/expand, zoom
- Clean API
 - Separation of concerns
 - Opportunity for clearly-defined contract
 - Readable domain-specific serialization format

Tools are interactive (and paper isn't)



Benefits of Domain Specific Languages

The good thing about bubbles and arrows,
as opposed to programs, is that they never
crash.

Bertrand Meyer



Is DSL new?

No, related topics are:

- “On the design and development of program families” David Parnas, 1976
- “Language Oriented Programming” Sergey Dimitriev, co-founder of JetBrains
- “Model Driven Architecture” by OMG
- “Language Workbenches” by Martin Fowler
- “Generative Programming: Methods, Tools and Applications” by Krzysztof Czarnecki and Ulrich Eisenecker.



Internal and External DSL

- Internal
 - The DSL is strongly related to the target languages; It contains specific syntax or constructs of the target language;
- External
 - Lack of symbolic integration;
 - The binding between the DSL and target language is weak; So errors in DSL can only be found if the compiler is made aware of the possible errors;

Source: Martin Fowler



Horizontal versus vertical DSL

- Horizontal DSL don't involve a particular industry, these are mainly focused on the technical aspects.
- Vertical DSL apply to a particular industry domain, these target the domain-expert of the business domain.

Source: "Domain-specific Development with Visual Studio DSL Tools" by Steve Cook and others



DSL Tools in Visual Studio

- Demo



Internal and External DSL

- Internal
 - The DSL is strongly related to the target languages; It contains specific syntax or constructs of the target language;
- External
 - Lack of symbolic integration;
 - The binding between the DSL and target language is weak; So errors in DSL can only be found if the compiler is made aware of the possible errors;

Source: Martin Fowler



Horizontal versus vertical DSL

- Horizontal DSL don't involve a particular industry, these are mainly focused on the technical aspects.
- Vertical DSL apply to a particular industry domain, these target the domain-expert of the business domain.

Source: "Domain-specific Development with Visual Studio DSL Tools" by Steve Cook and others

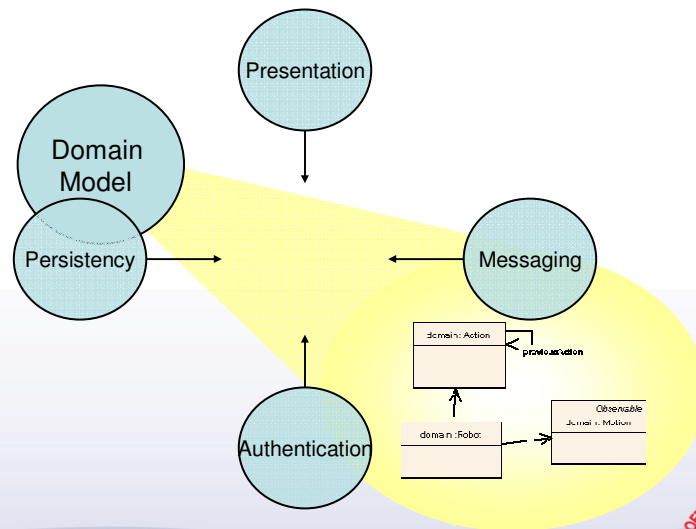


Bring DDD and DSL together

Where domains meet ...



DDD and Horizontal DSL:

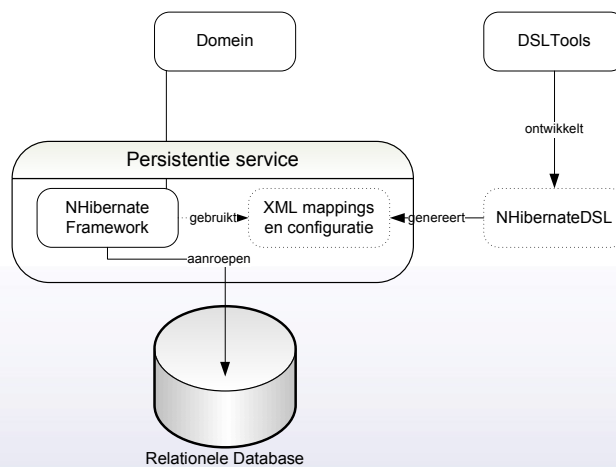


An example: DSL for persistency

Generating the NHibernate meta files



The objective for the DSL



Conclusion



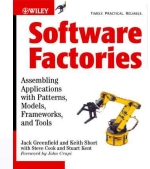
Conclusion

- Horizontal DSL can deliver much more productivity in a DDD development environment;
- Develop a DSL only if the work is repetitive;



Resources

- Domain-specific Development with Visual Studio DSL Tools
 - www.domainspecificdevelopment.com
- “Software factories”, Jack Greenfield
 - www.softwarefactories.com



Blogs:

- www.andreboonzaaijer.com
- www.robvens.nl/blog
- www.edwinvandillen.nl



Evaluation form

Vul je evaluatieformulier in en maak kans op een van de prachtige prijzen!!

Fill out your evaluation form and win one of the great prizes!!

Session Code: CT.01

