Domain Software Engineering

Prof. Dr. Frank J. Furrer

Ringvorlesung TU Dresden WS 2015/2016
Montag, 1. Februar 2016
16:40 – 18:10 / INF E006

V1.0 / 30.1.2016

Content

DSE

Introduction

Domain-Driven Design [DDD]
Domain Engineering [DE]
Domain-Specific Languages [DSL]
Domain Language Engineering [DLE]
Domain-Specific Modeling [DSM]

$\Rightarrow$

Domain Software Engineering [DSE]

DEFINITIONS

http://www.iconshut.com

Domain Software Engineering [DSE] =

an architectural ***methodology***

for evolving a *software* system

that closely aligns to *business* domains

Future-Proof Software-Systems: Domain Software Engineering

http://yaminfotech.org/services.html

There are different ways to approach software development.

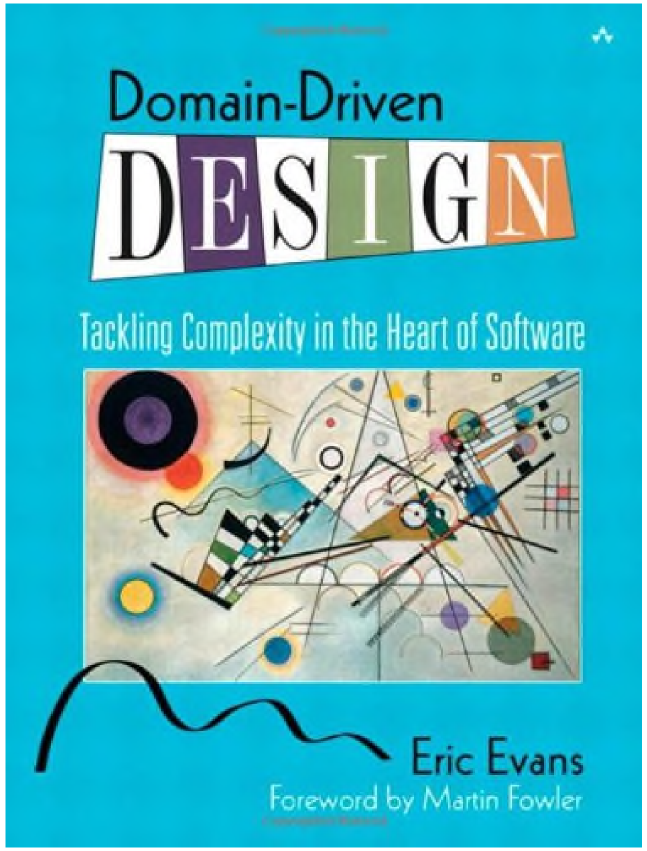
For the last 20 years, the software industry has known and used many methods to create its products – each with its advantages and shortcomings.

Here we investigate the method which started as «**Domain-Driven Design**».

2004:

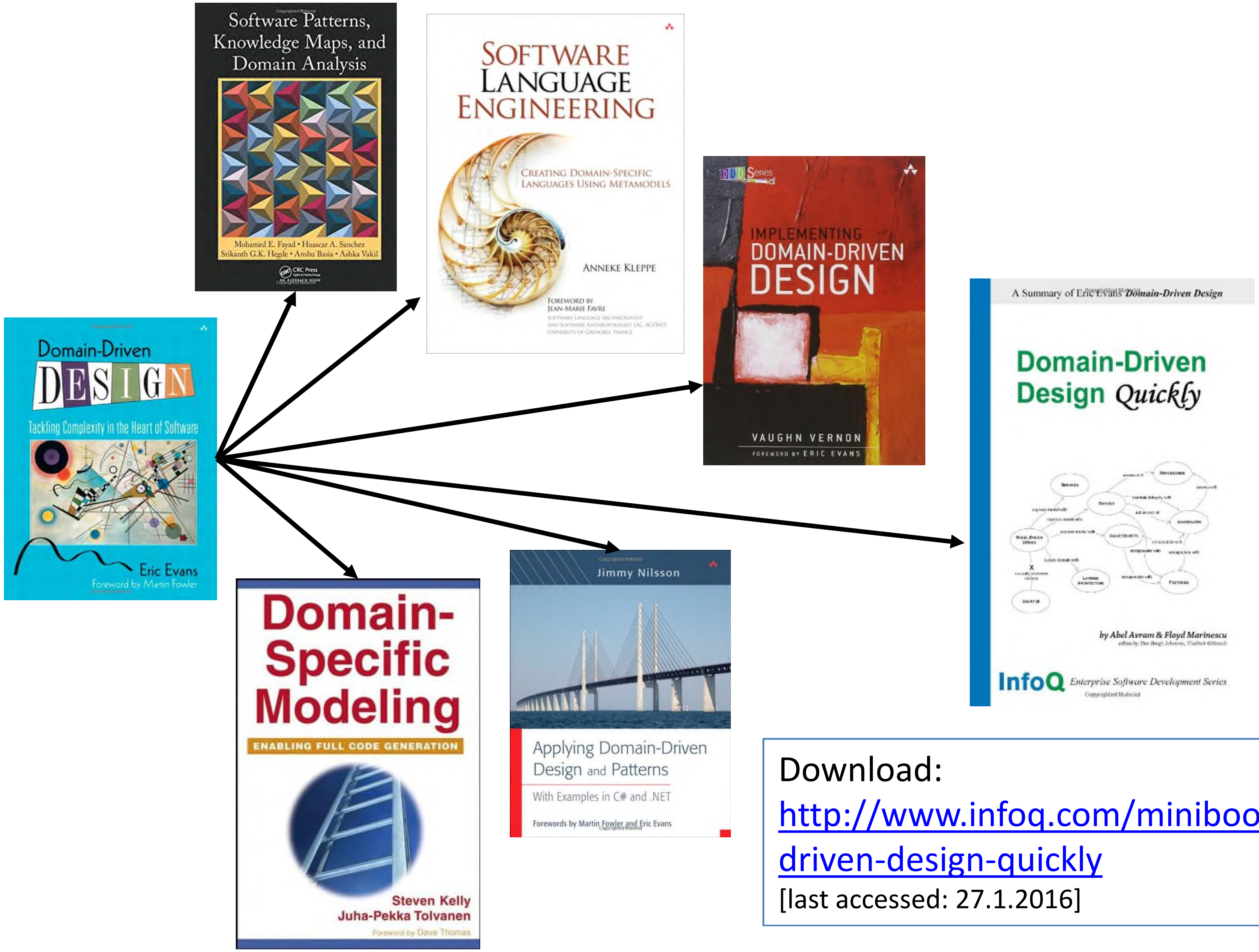
1990 … today:

Various Software Development Methodologies


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

ISBN 0-321-12521-5, 2004

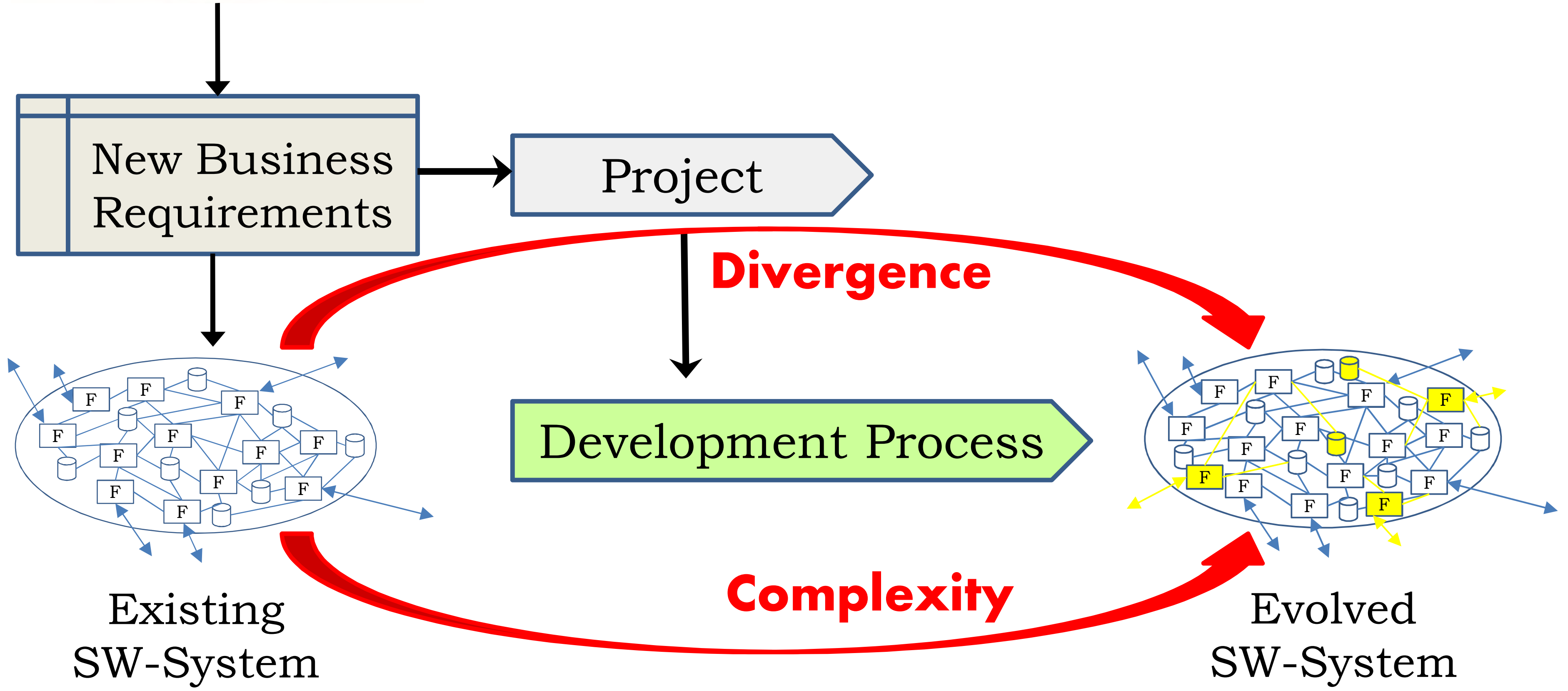
Ringvorlesung 2016: Domain Software Engineering


Software Patterns, Knowledge Maps, and Domain Analysis
Mohamed E. Fayad • Huascar A. Sanchez
Srikanth G.K. Hegde • Anshu Basia • Ashka Vakil
CRC Press
Software Language Engineering
Creating Domain-Specific Languages Using Metamodels
Anneke Kleppe
Foreword by Jean-Marie Favre
Implementing Domain-Driven Design
Vaughn Vernon
Foreword by Eric Evans
A Summary of Eric Evans' Domain-Driven Design
Domain-Driven Design Quickly
by Abel Avram & Floyd Marinescu
InfoQ Enterprise Software Development Series
Domain-Driven Design
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler
Domain-Specific Modeling
Enabling Full Code Generation
Steven Kelly
Juha-Pekka Tolvanen
Foreword by Dave Thomas
Jimmy Nilsson
Applying Domain-Driven Design and Patterns
With Examples in C# and .NET
Forewords by Martin Fowler and Eric Evans

Download:
http://www.infoq.com/minibooks/domain-driven-design-quickly
[last accessed: 27.1.2016]

DSE Context

http://www.clipartpanda.com


New Business Requirements
Project
Divergence
Development Process
Complexity
Existing SW-System
Evolved SW-System
F

http://www.3ilmchar3i.net

Divergence

Development Process

Complexity

http://www.fotosearch.com

Serious ***obstructions*** introduced by many software development processes

http://www.3ilmchar3i.net

DSE

Divergence

http://www.sutherlandweston.com

DEFINITIONS

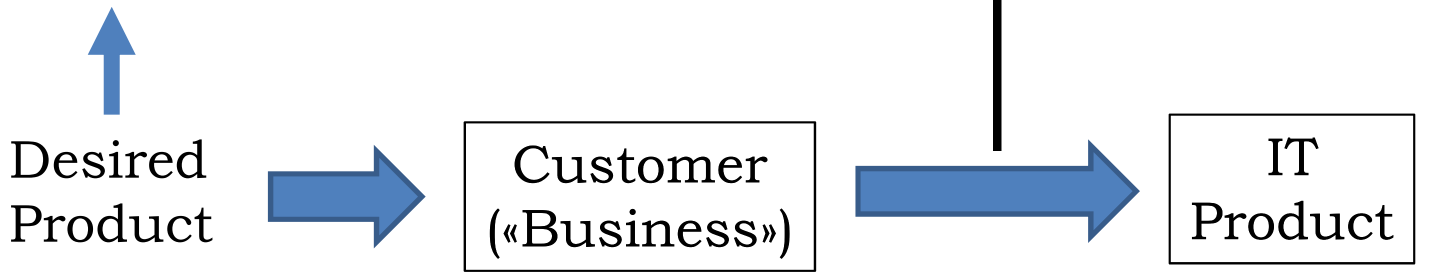
http://www.iconshut.com

Divergence =

Mismatch between *Business Needs*

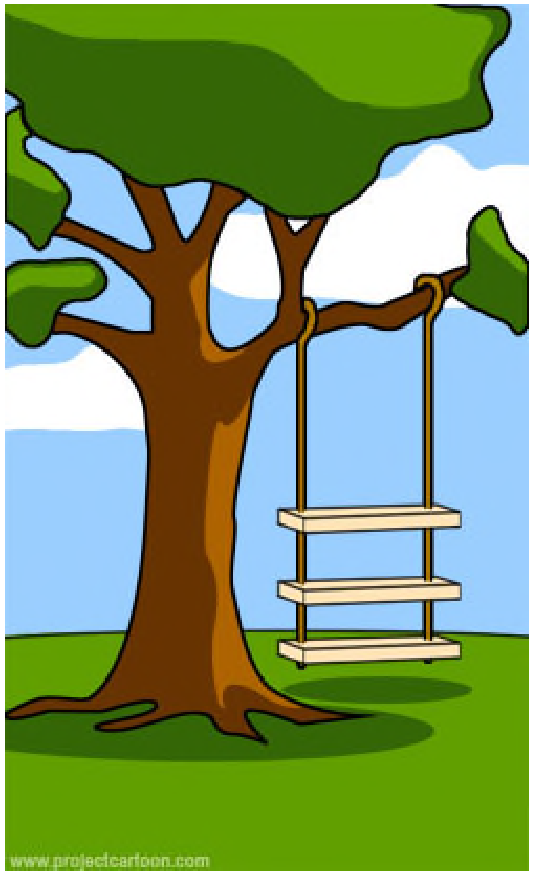
and *IT-Implementation*

Example: «Please build a swing for my little daugther»

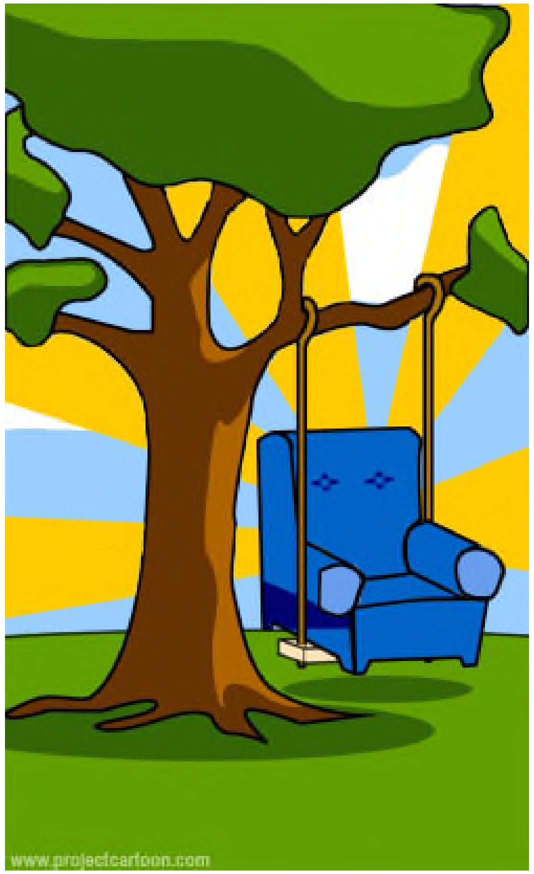


Requirements → Specifications →

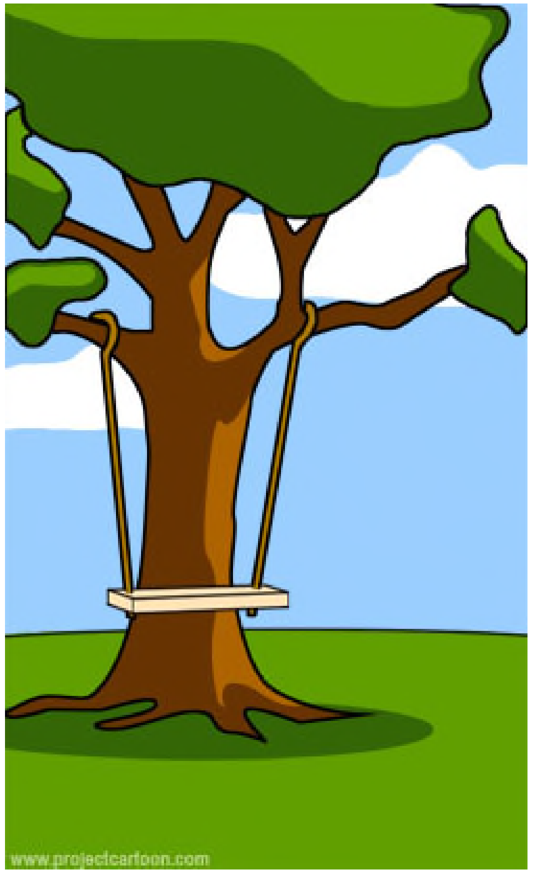
http://projectcartoon.com/create/



How the customer explained it



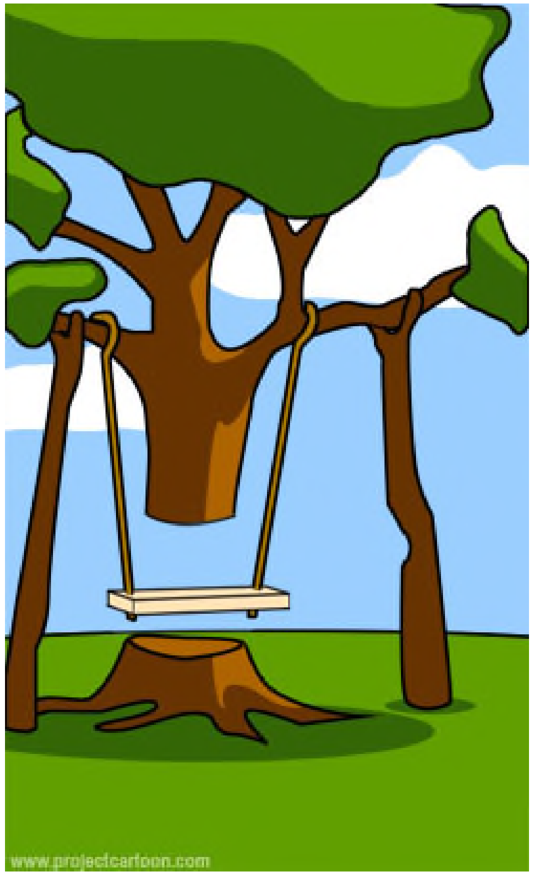
How the business consultant described it



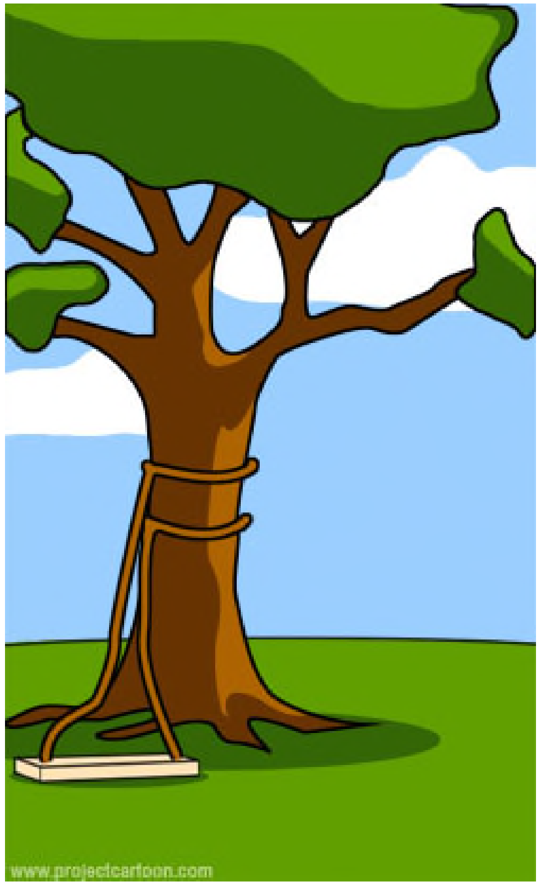
How the project leader understood it

Implementation

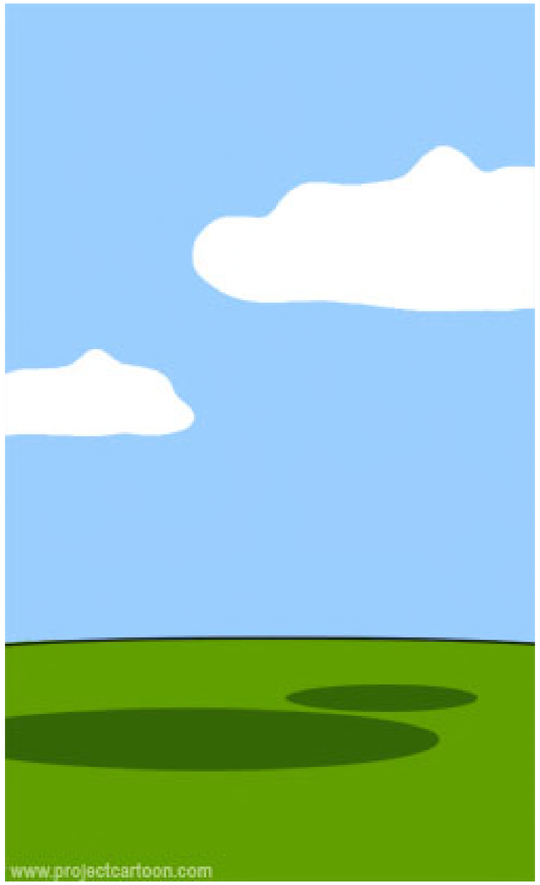
http://projectcartoon.com/create/



How the analyst designed it



How the programmers implemented it



How the project was documented

Deployment → Operation

www.projectcartoon.com

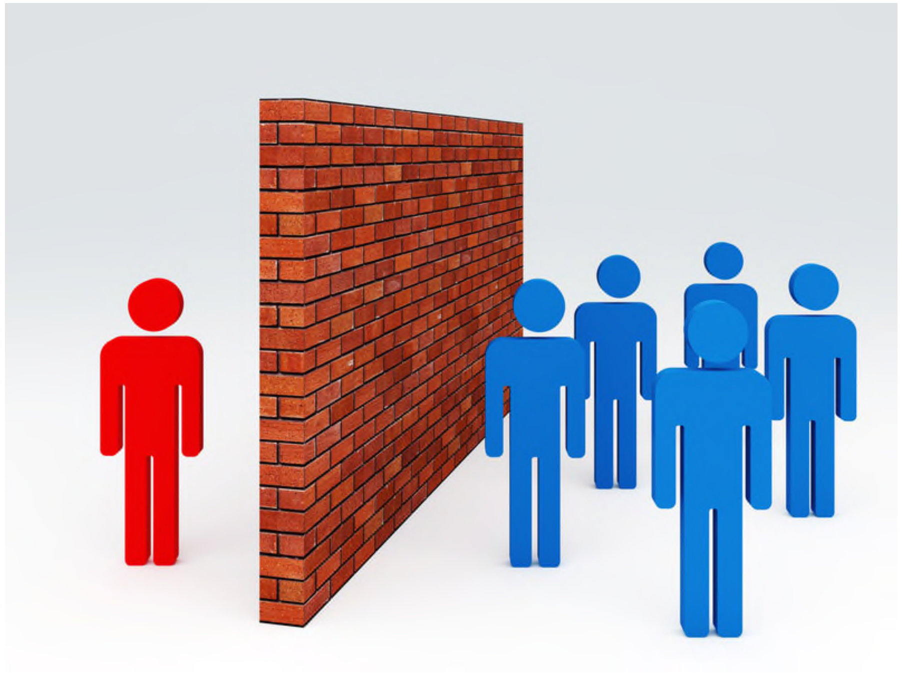
When the project was delivered

What the customer really wanted

http://projectcartoon.com/create/

What is the reason?

Failed Communications!



http://mayrsom.com

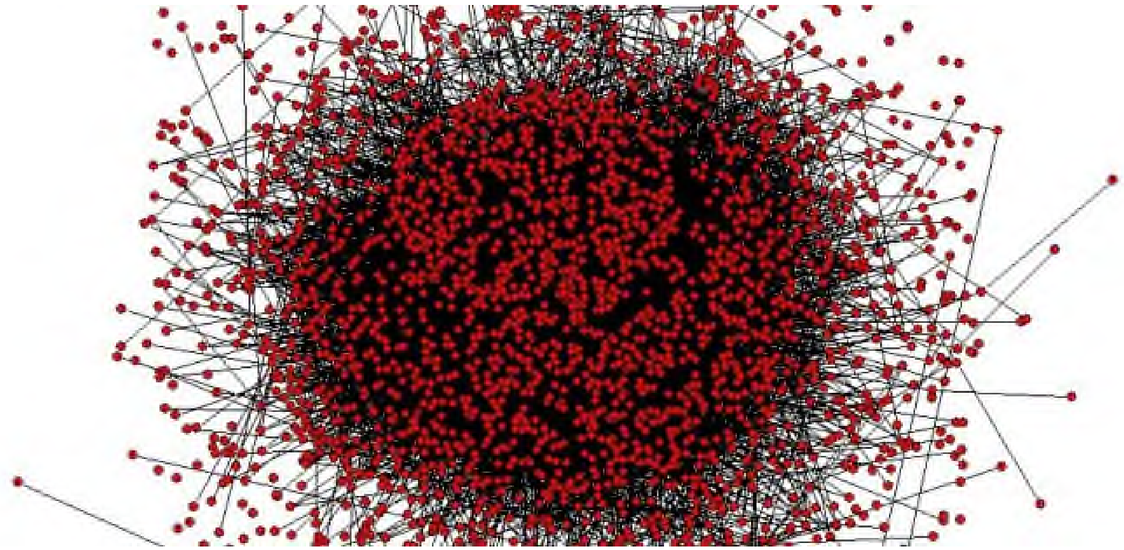
- Different *vocabulary* between business and IT
- Lots of *implicit* knowledge and assumptions
- No common *model*

http://www.fotosearch.com

DSE

Complexity

Complexity



http://blog.digital.telefonica.com

DEFINITIONS

“**Complexity** is that property of an IT-system which makes it difficult to formulate its overall behaviour, even when given *complete* information about its parts and their relationships“

Complexity = (IT-) Risk

$$\begin{tikzcd}
 & \boxed{\textbf{Essential} \text{ complexity}} \\
\boxed{\textbf{2} \text{ types of complexity}} \arrow[ur] \arrow[dr] & \\
 & \boxed{\textbf{Accidental} \text{ Complexity}}
\end{tikzcd}$$

http://www.acordceo.org

https://acoarecovery.wordpress.com

Essential complexity

… is the *inherent* complexity of the system to be built.

Essential complexity for a given problem *cannot* be reduced.

It can only be lessened by *simplifying* the requirements for the system extension.

Manage essential complexity

Accidental Complexity

… is *introduced* by our development activities or by constraints from our environment.

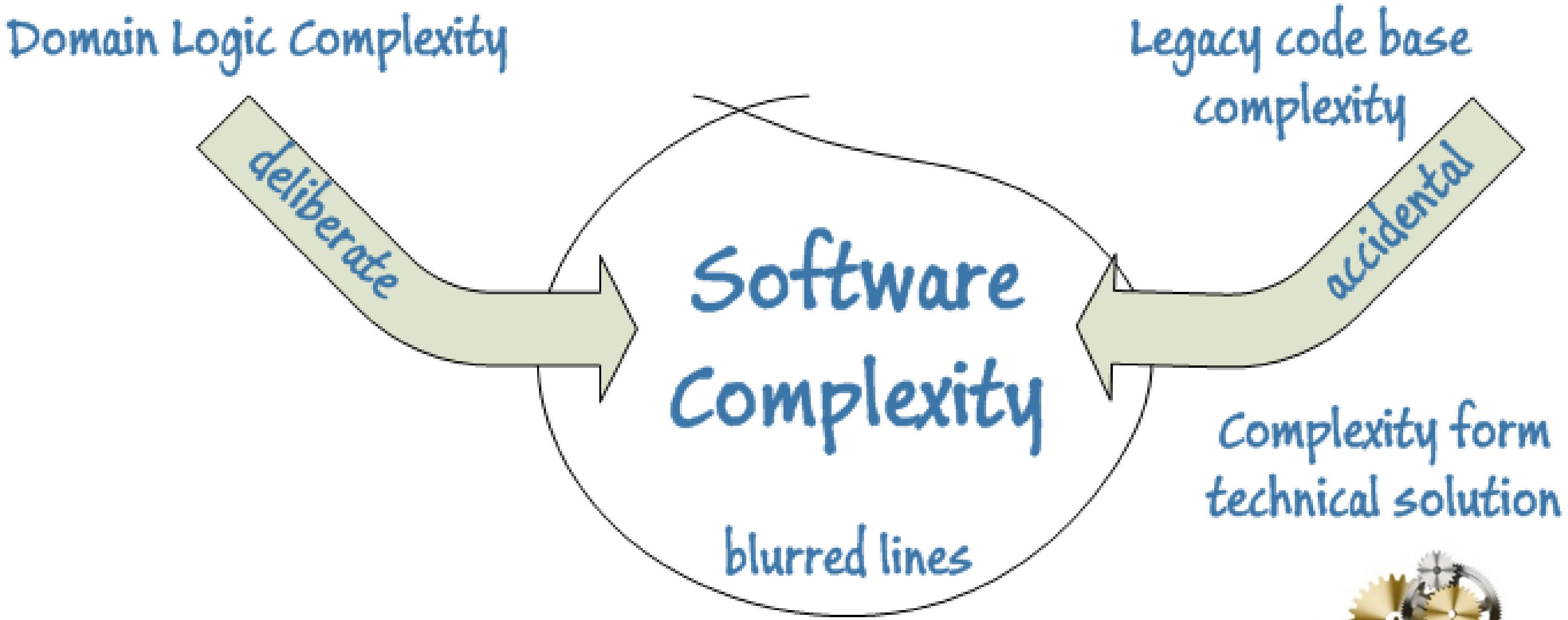
This is unnecessary and can be *reduced* or eliminated.

⇒ Development methodology!

Combat accidental complexity

Business Plan

pppst.com


Domain Logic Complexity
Legacy code base complexity
deliberate
accidental
Software Complexity
blurred lines
Complexity form technical solution

DSE

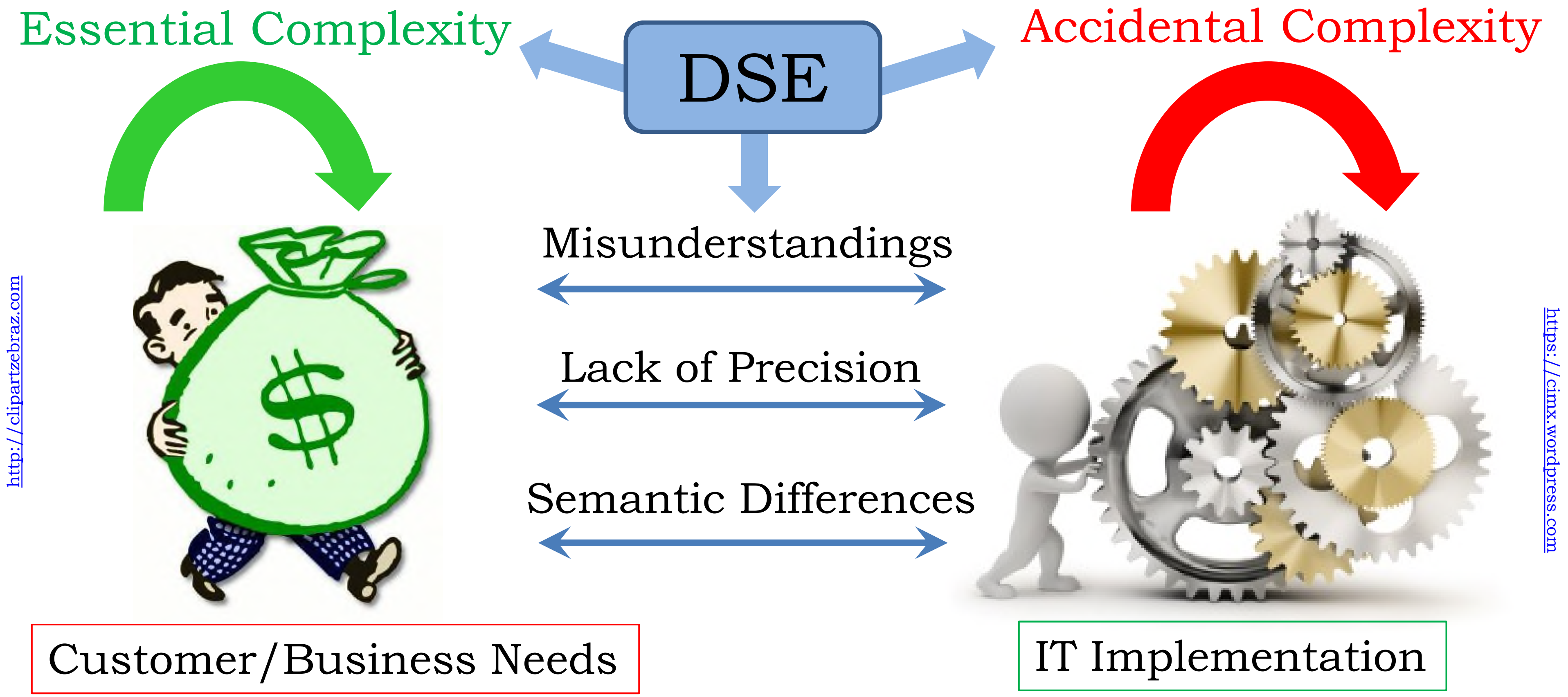
Fundamentals (1/2)

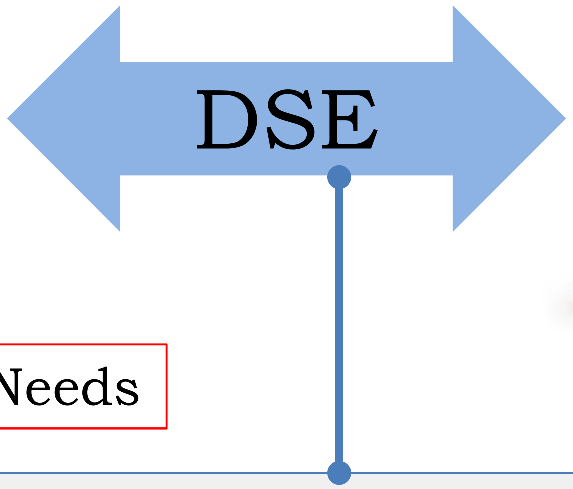
Frustration !

Divergence =

Mismatch between:

Business Needs ⇔ IT-Implementation





Customer/Business Needs

IT Implementation

Which are the key elements of DSE (Domain Software Engineering?)

1. Understanding the Business/Application Domain in terms of the business
   (⇒ Domain Model)
2. Use of an ubiquitous language
   (Business ⇔ IT alignment)
3. Software: Implementation of Business Domain concepts
   (Concepts ⇒ Business objects ⇒ Programm objects)

Universale Ausdrucksform

The DSE concepts:

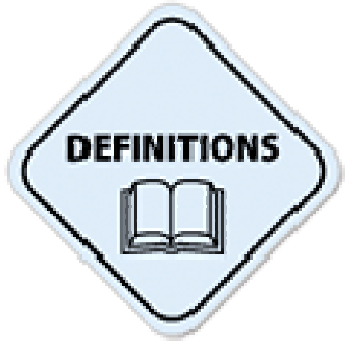
Business/Application Domain

Bounded Context

Domain Model

Anticorruption Layer

http://blogs.msdn.com



http://www.iconshut.com

Business/Application Domain =

A Domain is a Sphere of Knowledge, Influence or Activity.

A Domain lives within a Bounded Context.

A Domain represents a well-defined Part of the Real World.

A Domain encapsulates a Domain Model.

www.thinkddd.com

https://thoughtsfromthisflower.wordpress.com

http://www.iconshut.com

Bounded Context =

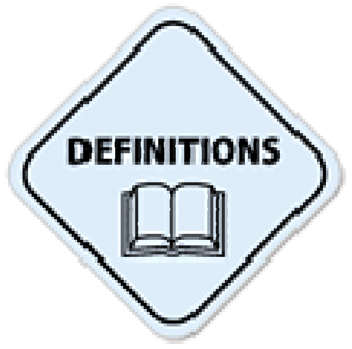
The Bounded Context is the Boundary of a Model.

When you have multiple Models you should define Bounded Contexts.

To map between Bounded Contexts you use a Context Map.

www.thinkddd.com

http://cliparts.co/meeting-pictures



http://www.iconshut.com

Domain Model =

A Domain Model is a representation of the Entities, Relationships and their Properties in your Domain

The Domain Model should be recognizable and understandable by the business *and* IT

The domain model has sufficient essential details

www.thinkddd.com

http://www.iconshut.com

Anticorruption Layer =

An Anti-Corruption Layer is a method to isolate two domains or systems, allowing systems to be integrated without knowledge of each other

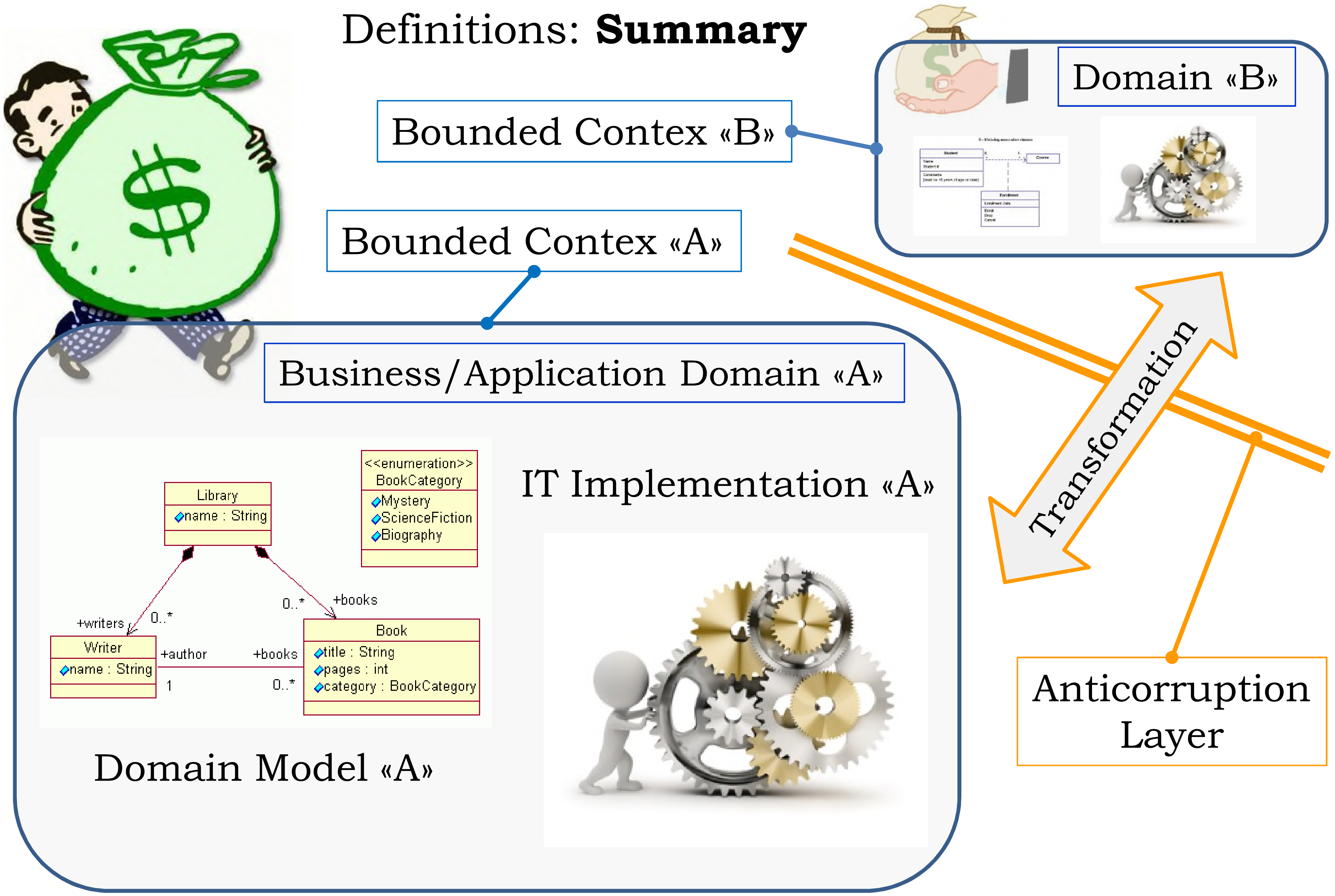
An Anti-Corruption Layer presents a Facade to both systems, defined in terms of their specific models

Anti-Corruption Layers maintain the integrity of differing systems and models

www.thinkddd.com

Definitions: **Summary**

http://clipartzebraz.com



Examples:

Business/Application Domain

Bounded Context

Domain Model

Anticorruption Layer

Example: Business/Application Domain

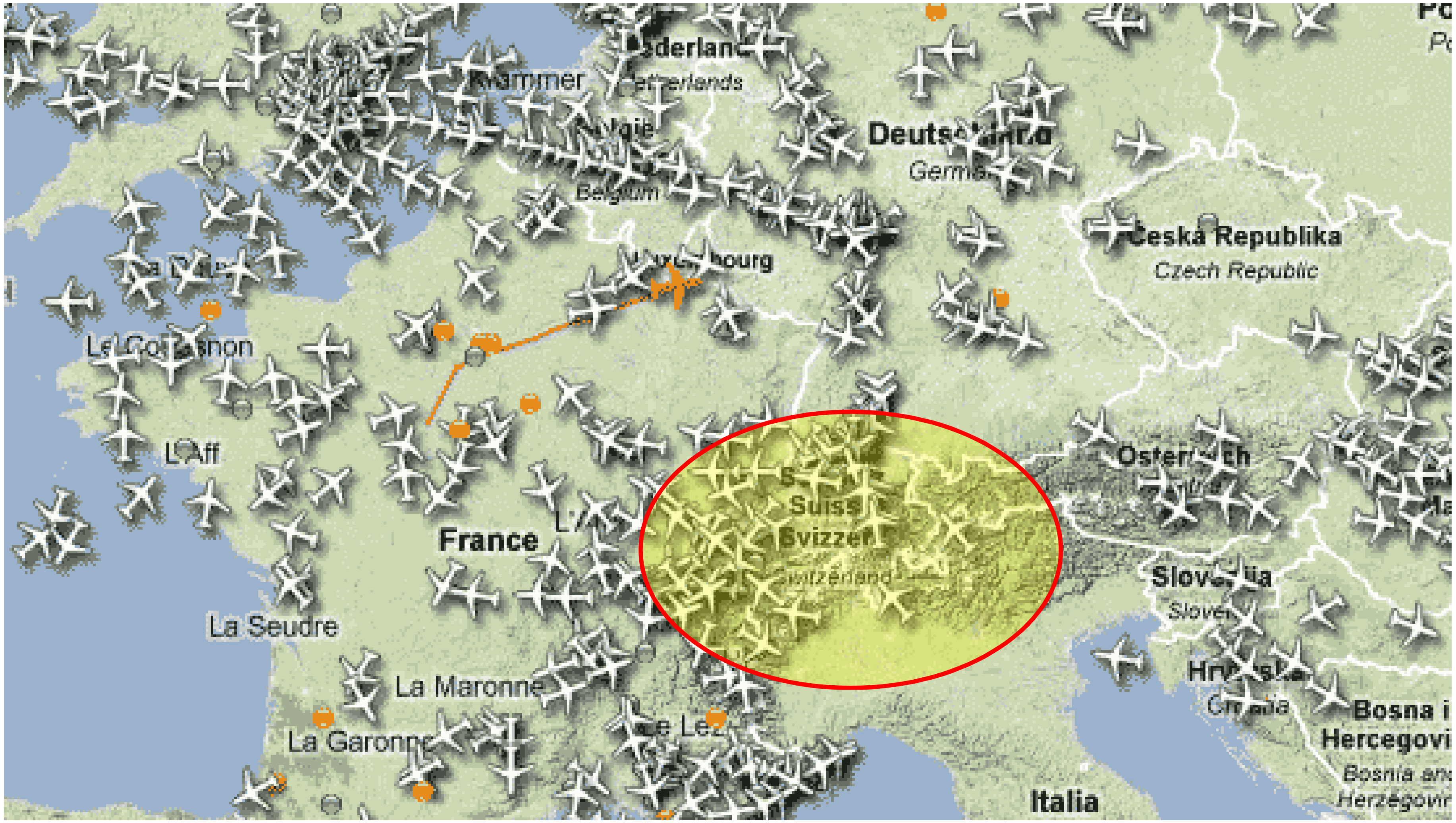
http://www.skyguide.ch

Domain = Flight Monitoring

Context:

Thousands of planes are in the air all over the planet. The flight monitoring systems track every flight and avoid collisions

Example: Bounded Context ⇔ SKYGUIDE Switzerland

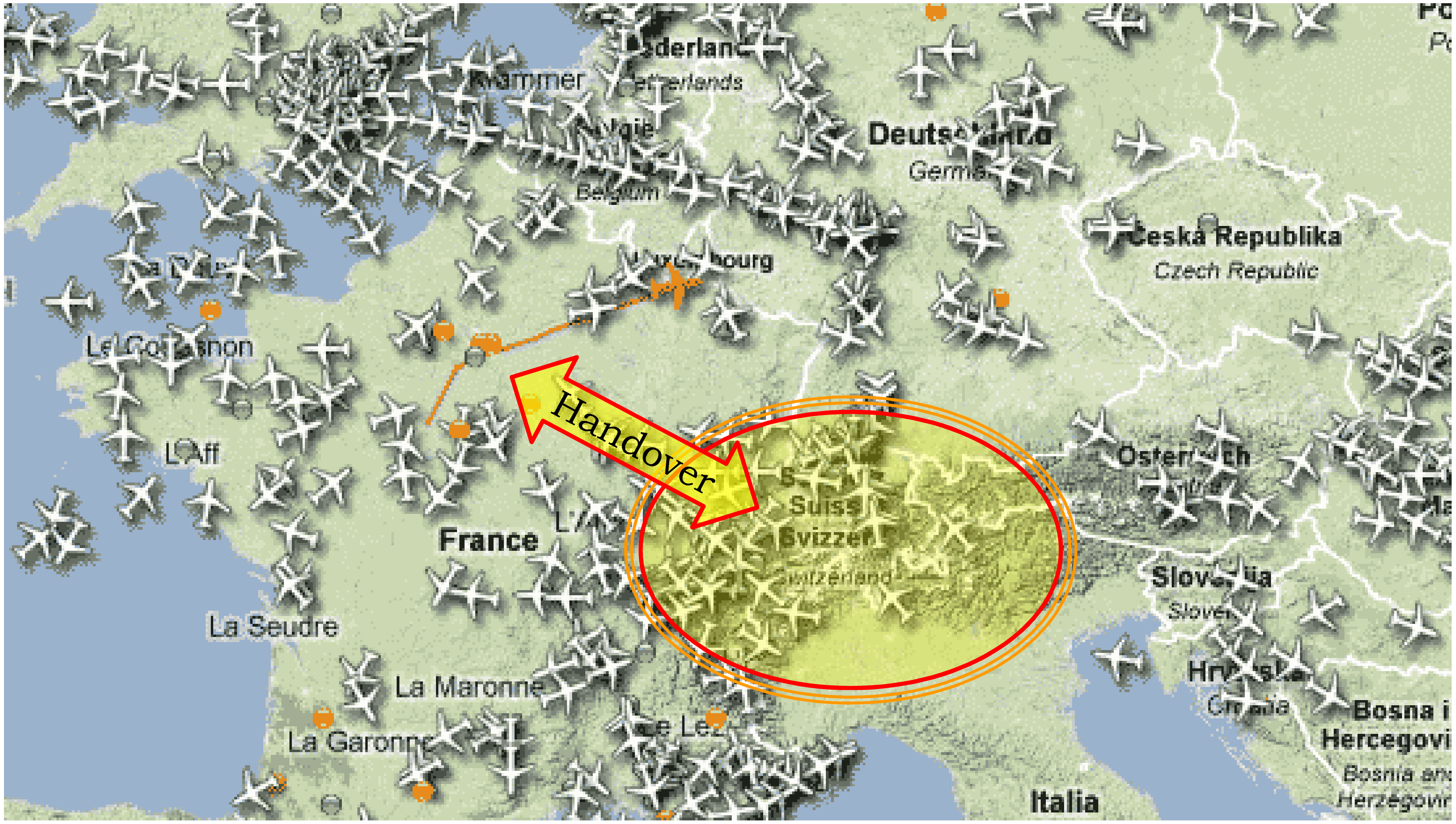


https://www.flightradar24.com

Boundary = Contractual Responsibility within the European System

Example: Bounded Context

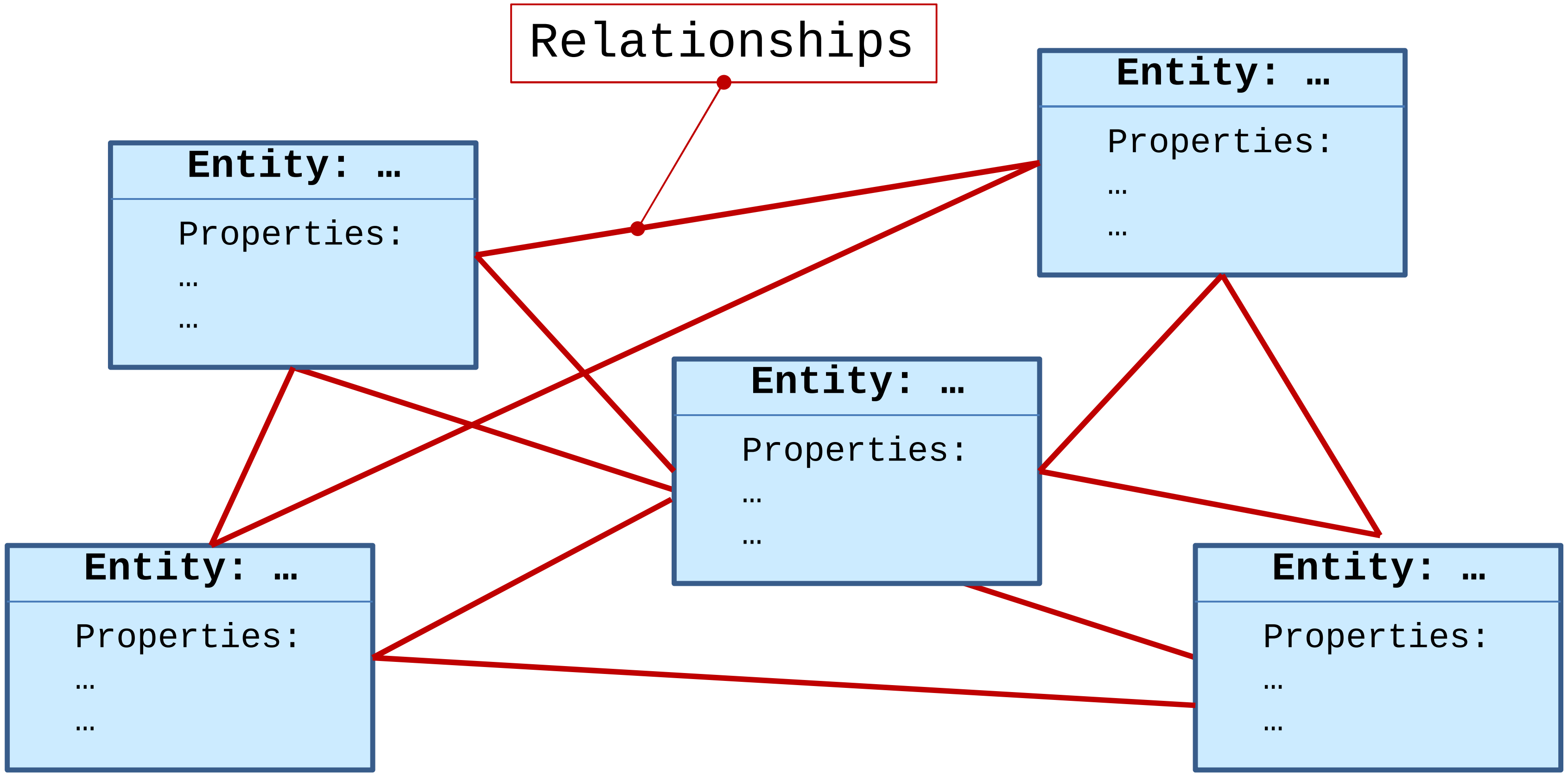
https://www.flightradar24.com



Anticorruption Layer = X-Compatibility Layer

Domain Model =

Reminder: A Domain Model is a representation of the Entities, Relationships and their Properties in your Domain

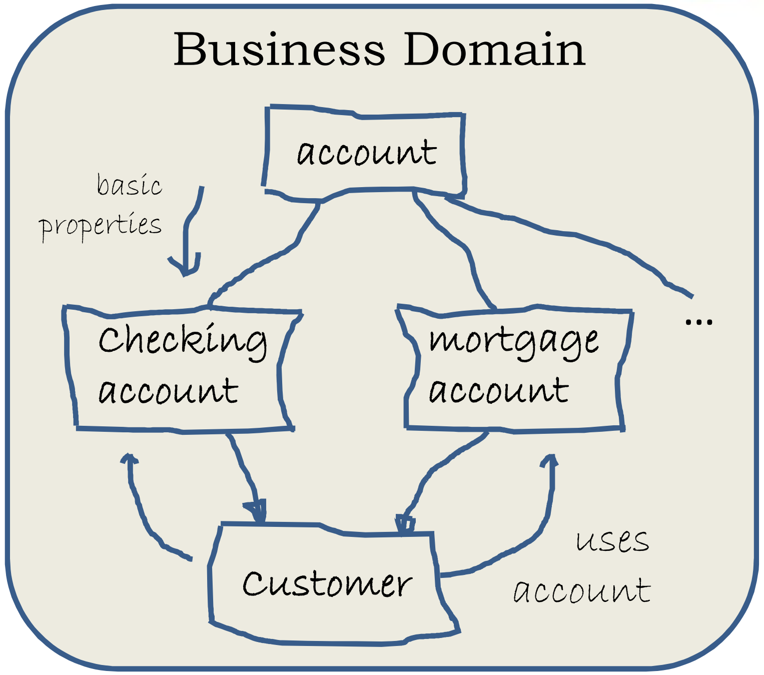


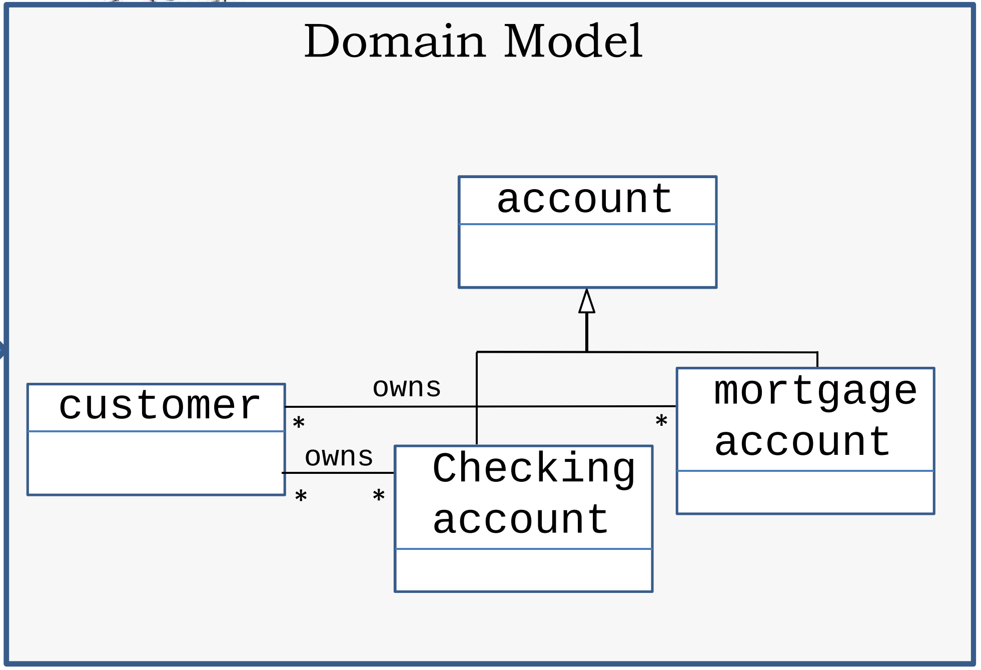
Dialog

http://www.faire-schule.ch

Domain Expert

IT Expert





http://www.faire-schule.ch

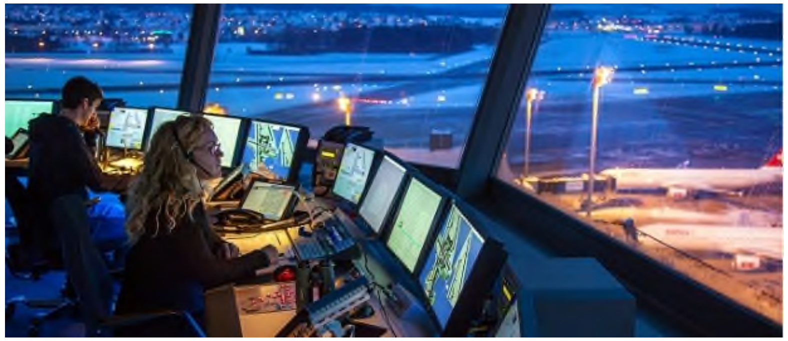
Domain Expert

IT Expert

«A domain model is not just the knowledge in a domain expert's head -
… it is a rigorously organized and selective abstraction of that knowledge»

Eric Evans, 2004

Example 1:
Flight Monitoring **Domain Model**

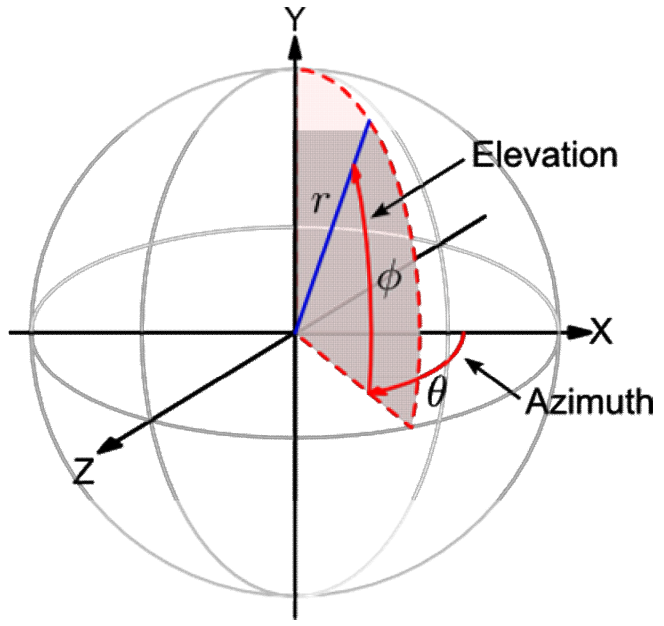


… Development of the Domain Model
⇒ Search & Definition of **Key Concepts**

$$\begin{tikzcd} & & \text{Departure} \\ \text{Aircraft} \arrow[r] & \text{Route} \arrow[ur] \arrow[dr] & \\ & & \text{Destination} \end{tikzcd}$$

Example 1: Flight Monitoring **Domain Model**

$$\begin{tikzcd}
\text{Aircraft} \arrow[r] & \text{Route} \arrow[r] & \text{Fix} \arrow[d] \\
 & & \text{3D-Point}
\end{tikzcd}$$

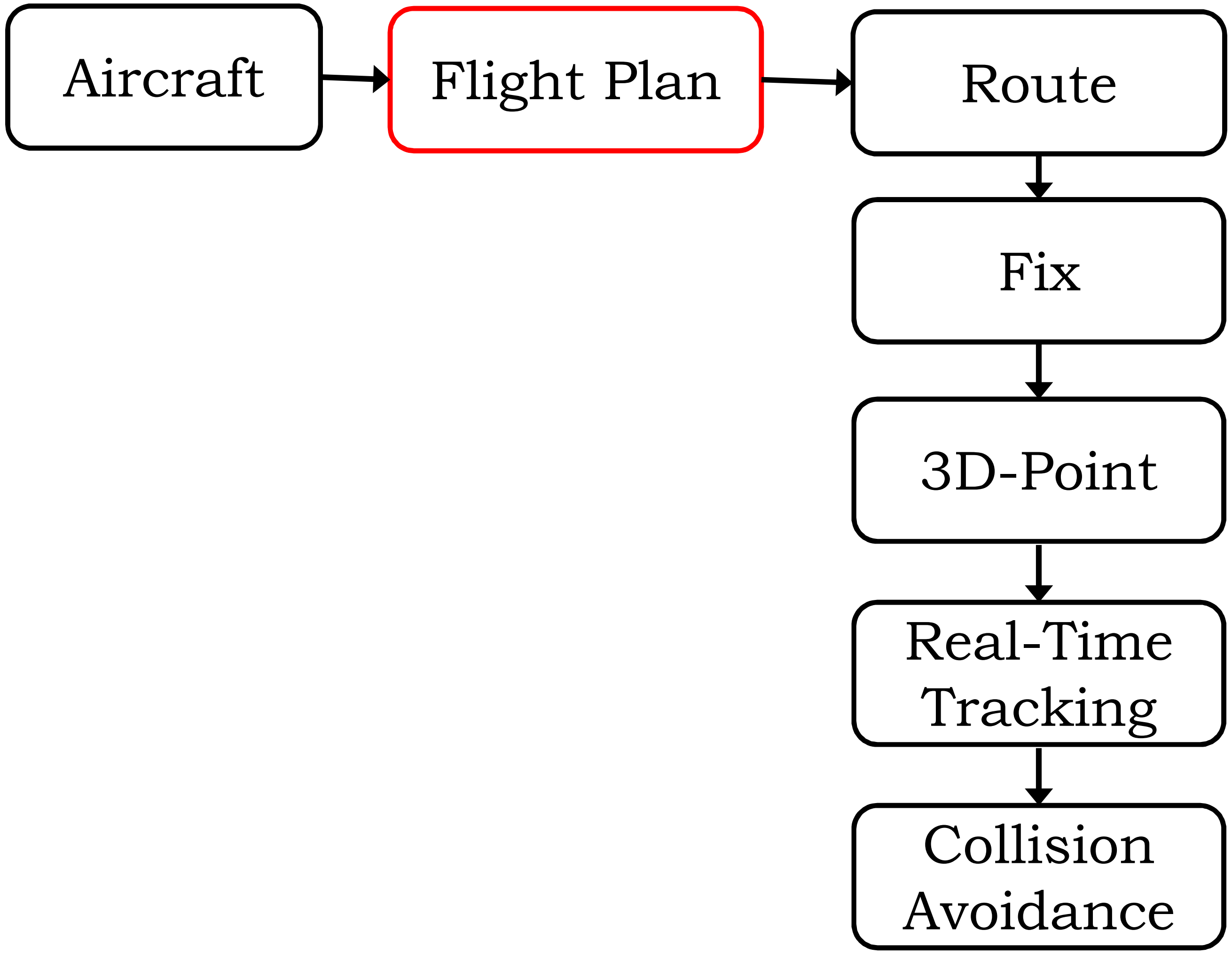


Example 1:
Flight Monitoring **Domain Model**

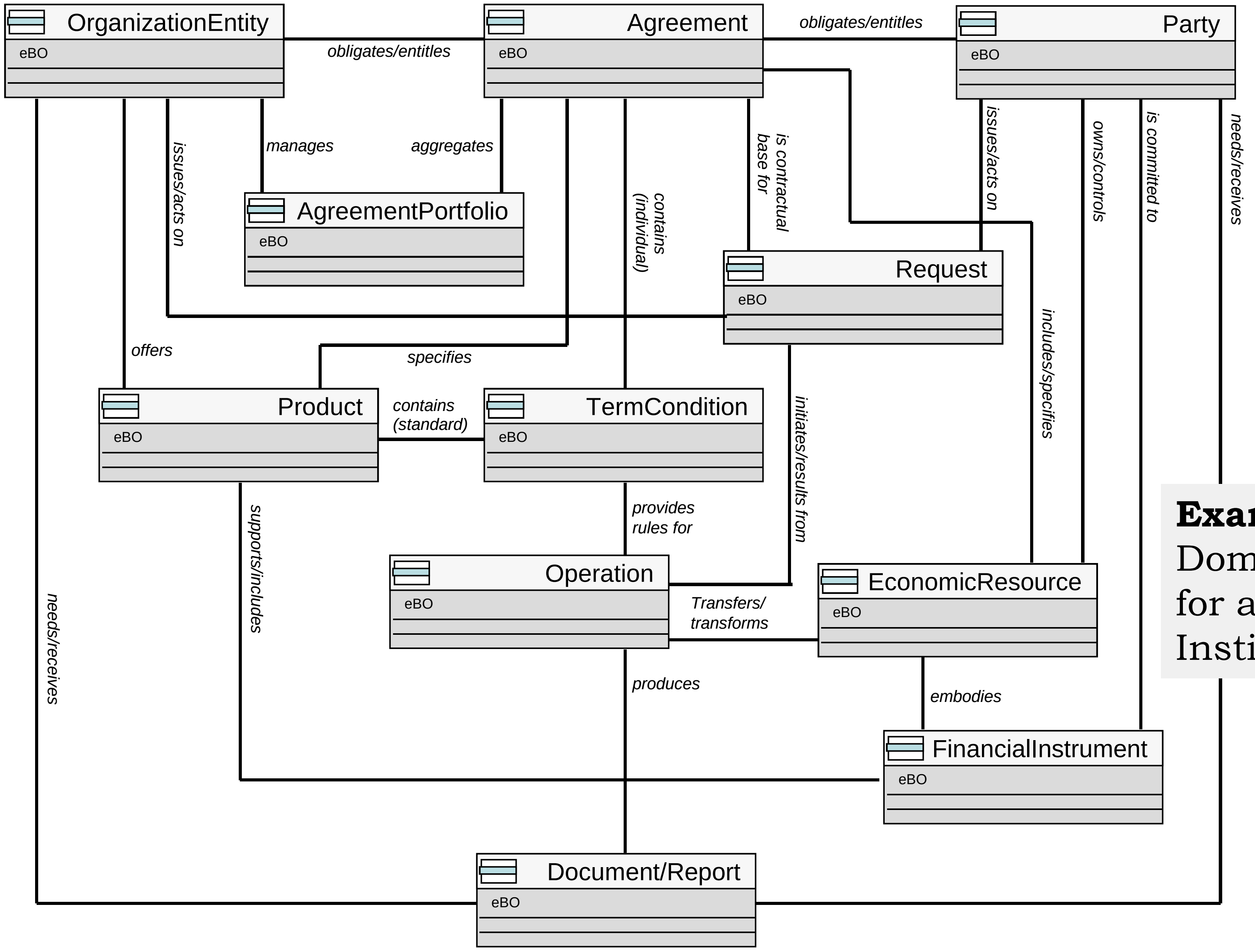
$$\begin{tikzcd}
\boxed{\text{Aircraft}} \arrow[r] & \boxed{\text{Flight Plan}} \arrow[r] & \boxed{\text{Route}} \arrow[d] \\
 & & \boxed{\text{Fix}} \arrow[d] \\
 & & \boxed{\text{3D-Point}}
\end{tikzcd}$$

Ohio
Columbus
Newark
Pittsburgh
Johnstown
Harrisburg
Dayton
South Columbus
Zanesville
Wheeling
Linn Run State Park
York
Kettering
Hamilton
Marietta
Morgantown
Cumberland
Maryland
Cincinnati
Parkersburg
Winchester
Columbia
Bethesda
Washington
West Virginia
Summit
Charleston
Webster Springs
Harrisonburg
Lexington
Nicholasville
Charlottesville
Palmyra
Milford

Example 1:
Flight Monitoring **Domain Model**



High-Level Domain Model



Example 2: Domain *Model* for a Financial Institution

http://knowhow.visual-paradigm.com

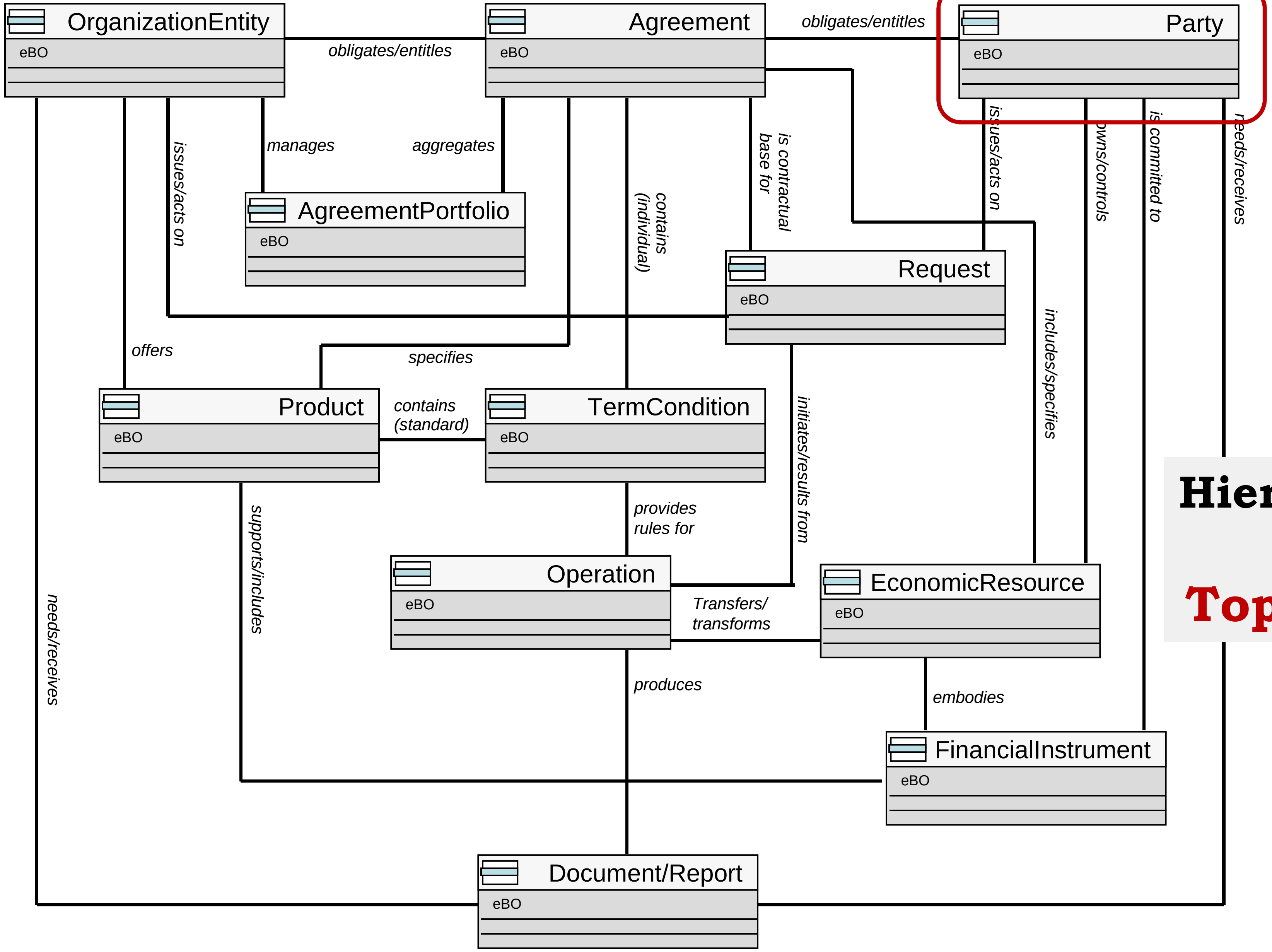
Problem:

Model-Explosion.

⇒ Size of the models grows!

Build hierarchical models

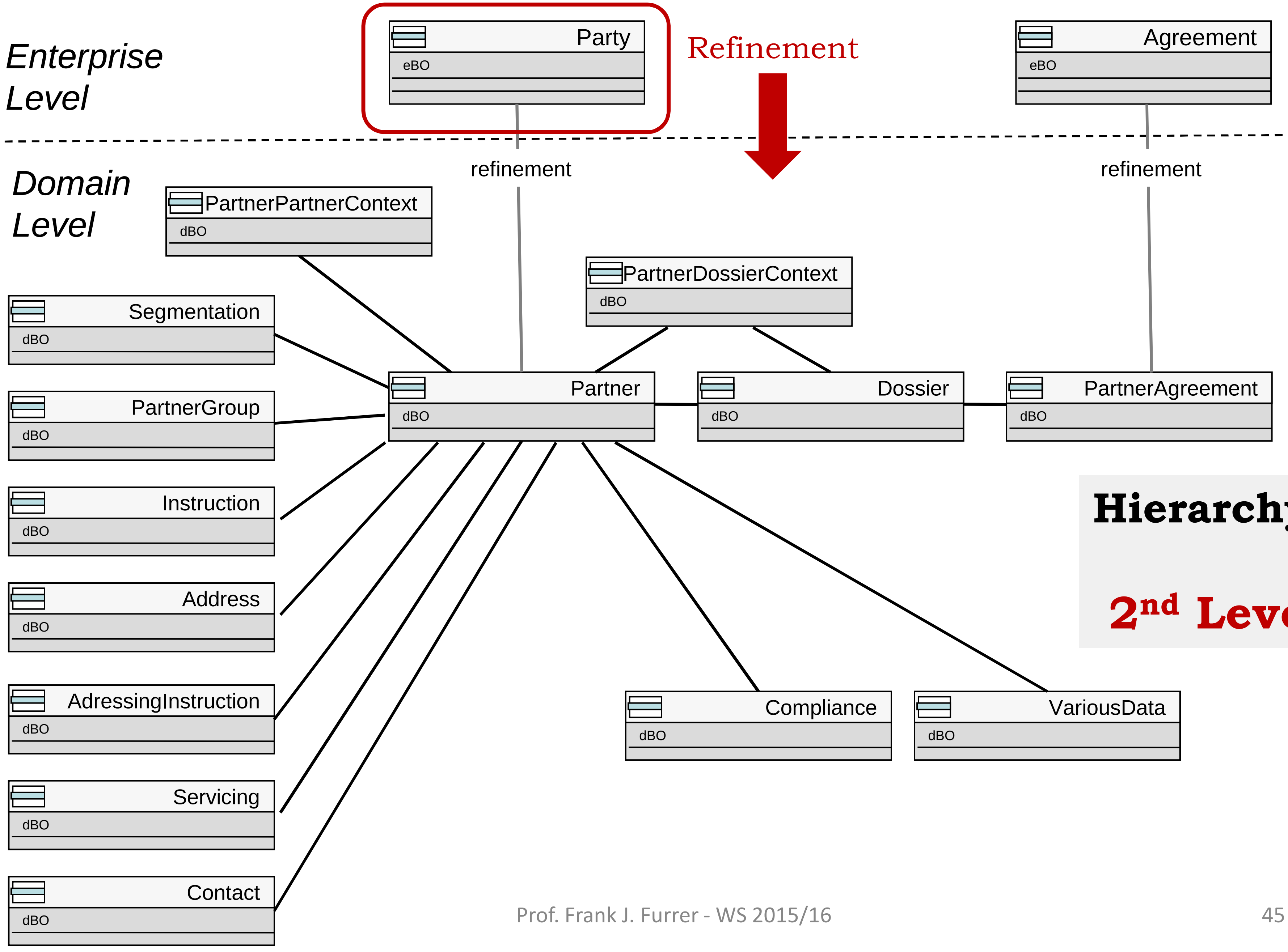
Ringvorlesung 2016: Domain Software Engineering



Hierarchy:

Top Level

Ringvorlesung 2016: Domain Software Engineering



Hierarchy:

2nd Level

Building a successful Domain Model

http://www.faire-schule.ch

Domain Expert

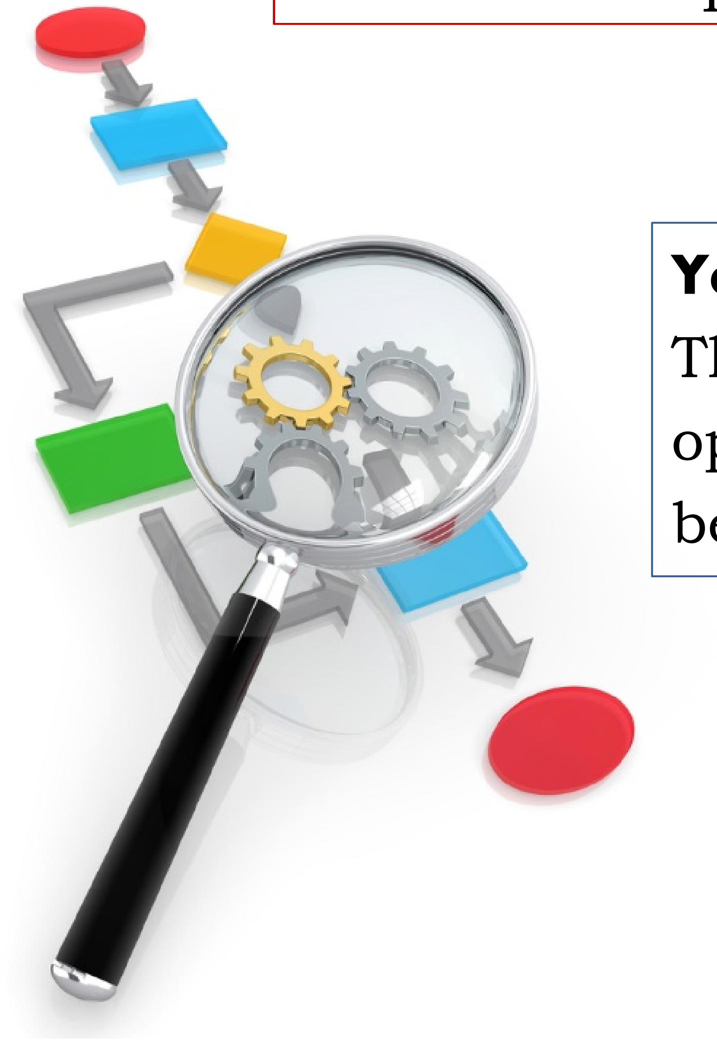
IT Expert

Fair, constructive and open dialog

Identify and describe:
- Business concepts in the domain
- Relationships
- Attributes and contraints

Is behaviour part of a Domain Model?

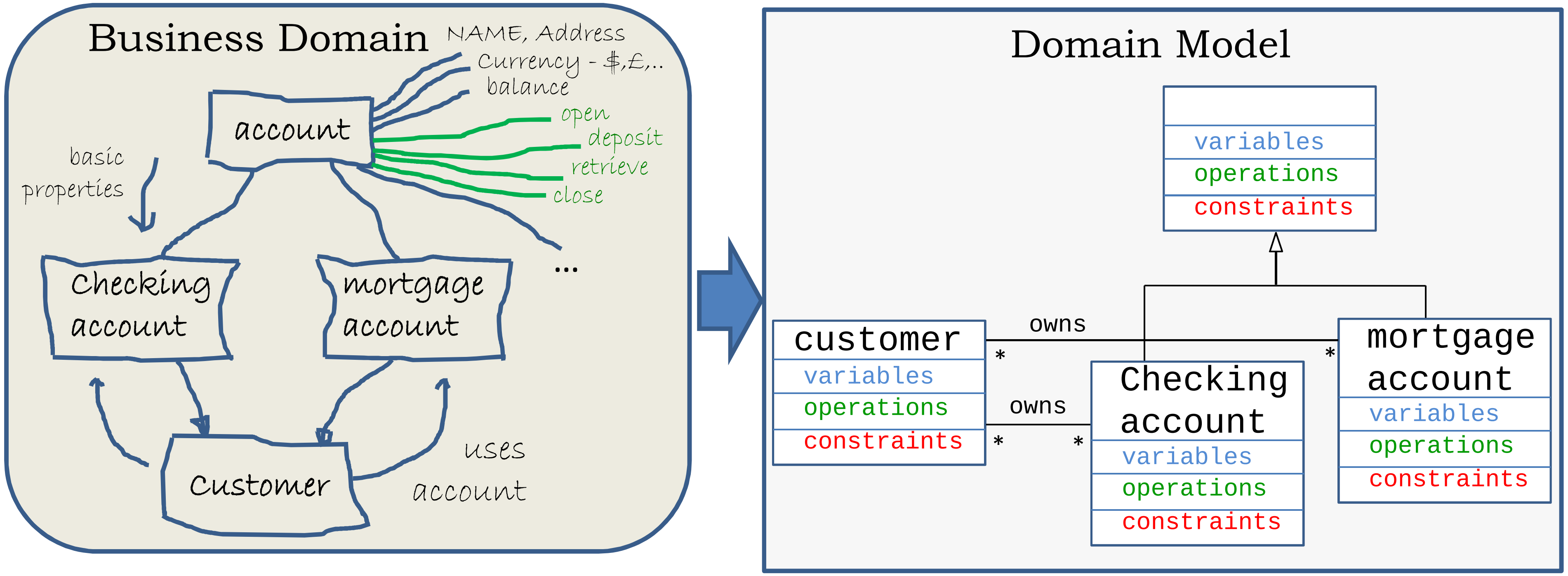
http://www.healthandlife.com.au



Yes!

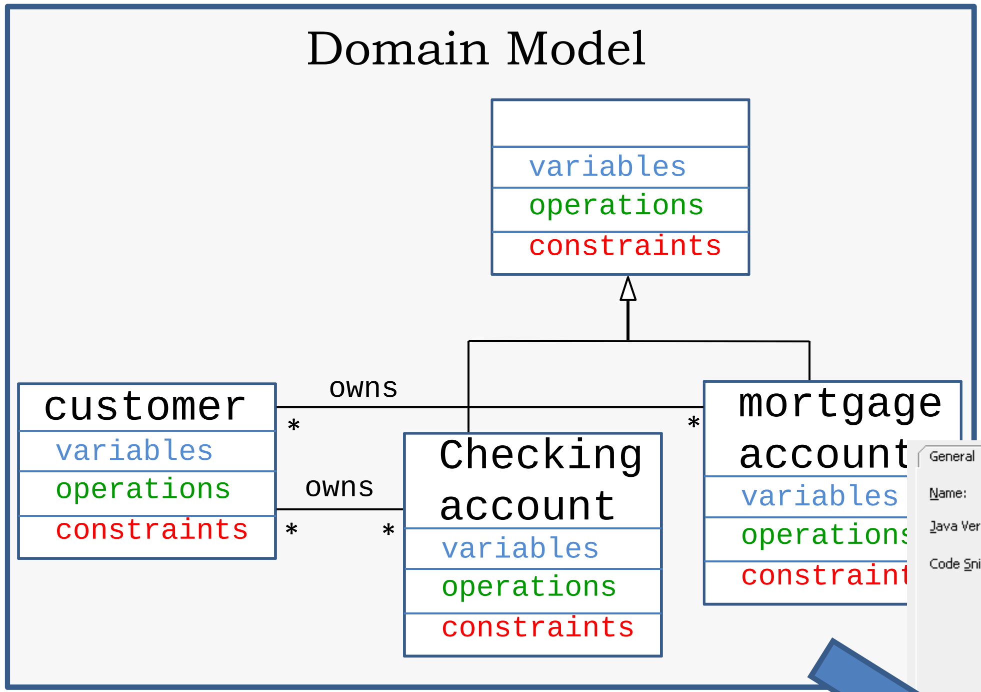
The variables, constraints and operations on domain concepts must be identified and specified

Is behaviour part of a Domain Model?

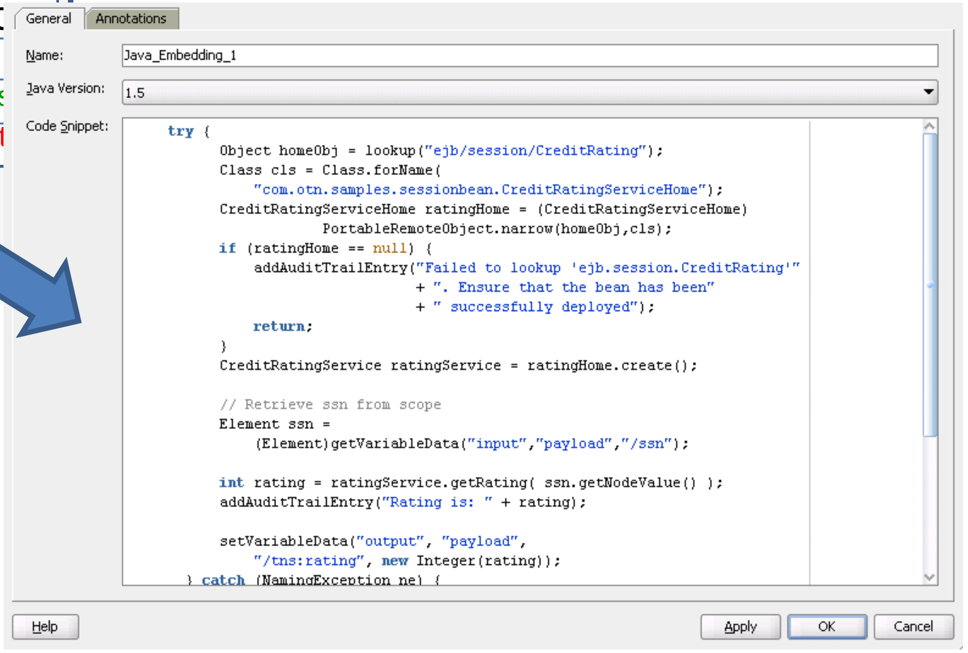


⇒ Model enrichment

Model ⇒ Code



«Having created a great model, but failing to properly transfer it into code will end up in software of questionable quality»



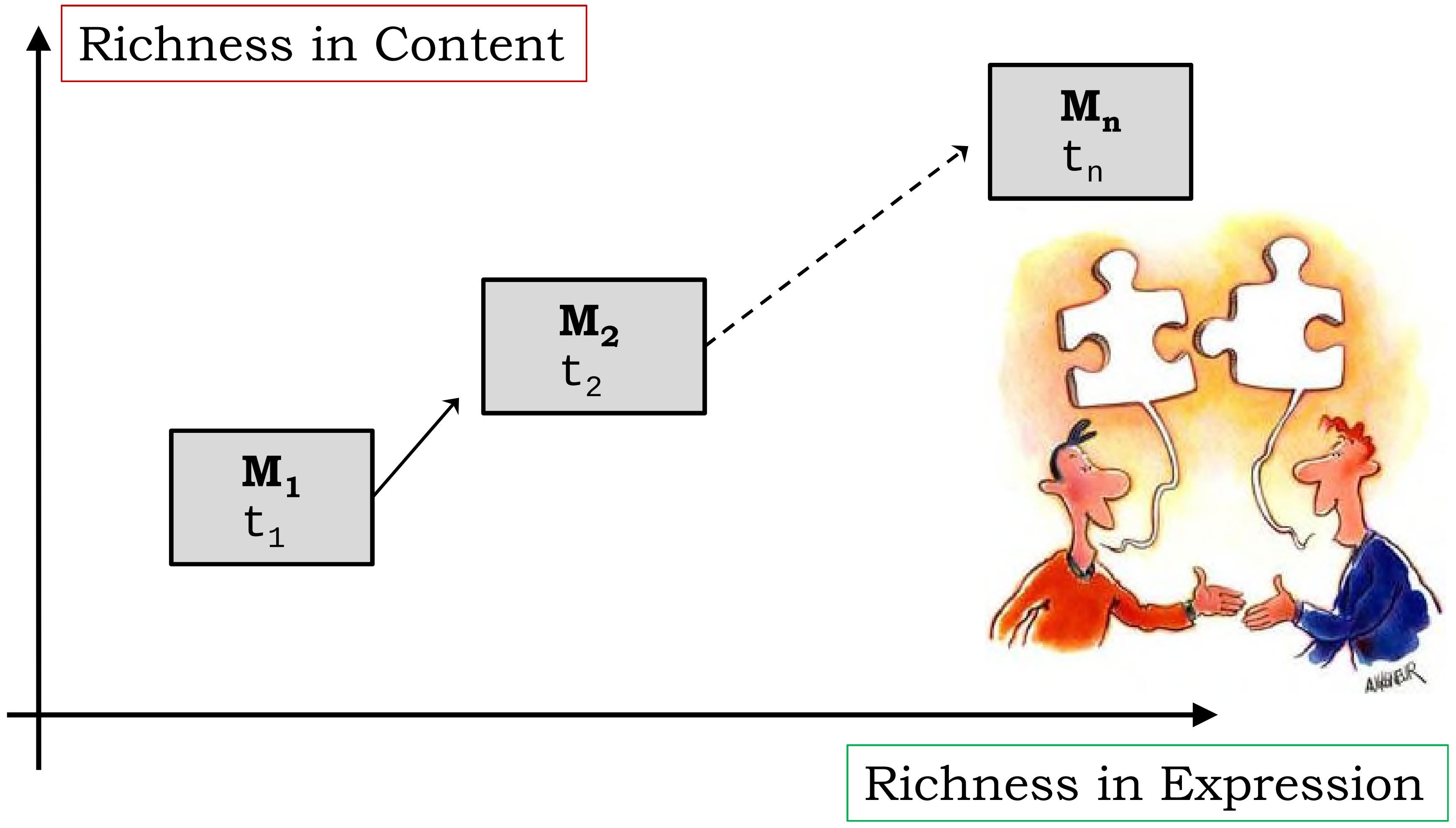
Name: Java_Embedding_1

Java Version: 1.5

Code Snippet:

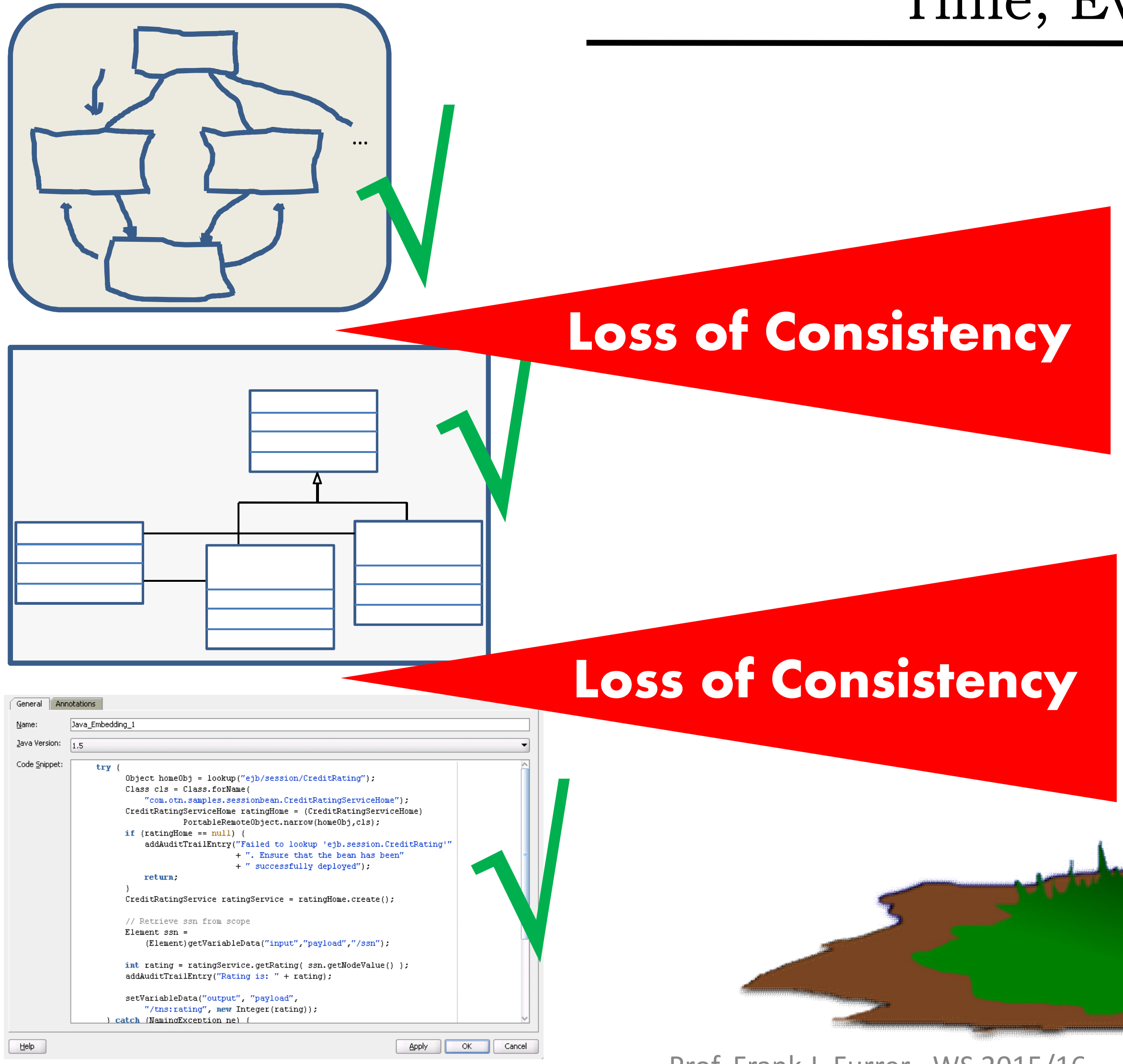
```
    try {
        Object homeObj = lookup("ejb/session/CreditRating");
        Class cls = Class.forName(
            "com.otn.samples.sessionbean.CreditRatingServiceHome");
        CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
                    PortableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                             + ". Ensure that the bean has been"
                             + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

Continuous Evolution of the Domain Model



DSE: Loss of Consistency

Time, Evolution



General | Annotations

Name: Java_Embedding_1

Java Version: 1.5

Code Snippet:

```
    try {
        Object homeObj = lookup("ejb/session/CreditRating");
        Class cls = Class.forName(
            "com.otn.samples.sessionbean.CreditRatingServiceHome");
        CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
                    PortableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                              + ". Ensure that the bean has been"
                              + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

Help | Apply | OK | Cancel

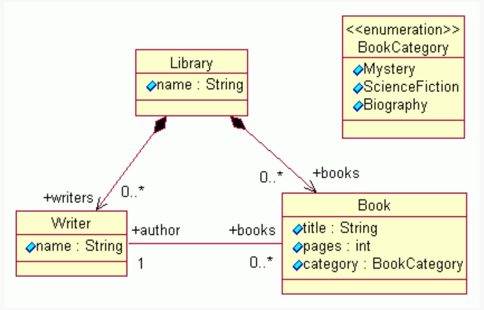
The code must be an expression of the model

A change in the code may need a change in the model

Anticorruption Layer =

Anti-Corruption Layers maintain the **integrity** of differing systems and models

Domain or System «**A**»



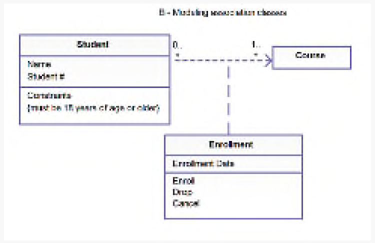
Concepts & Models

Semantic Mismatch

Concept inconsistency

etc.

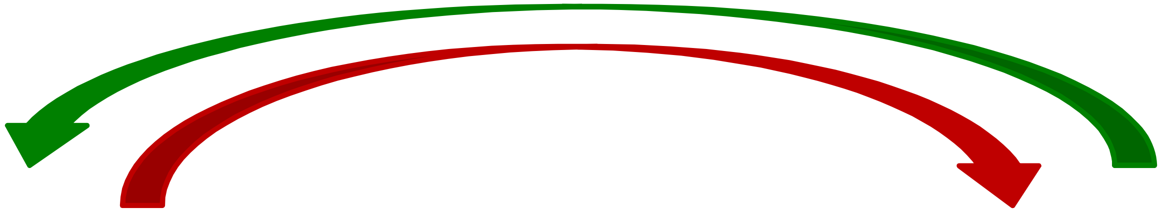
Domain or System «**B**»



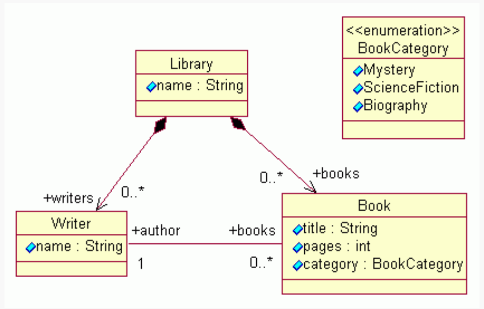
Concepts & Models

Anticorruption Layer

Explicit Mapping between contexts and code

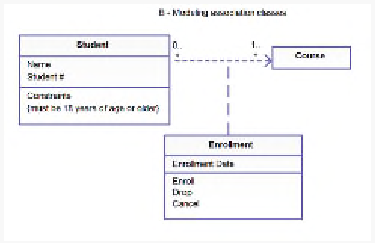


Domain or System «**A**»



Concepts & Models

Domain or System «**B**»

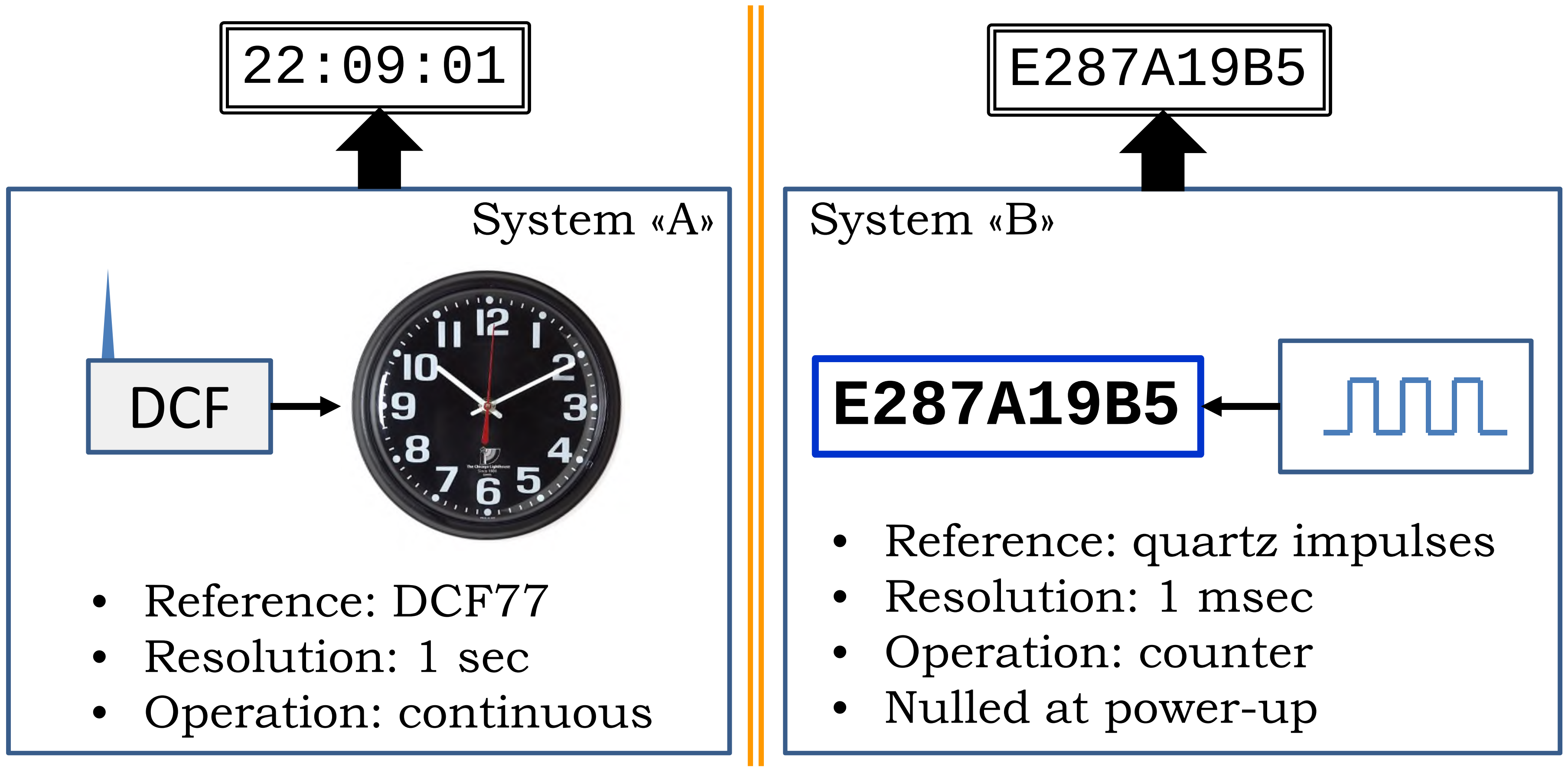


Concepts & Models

Example: Concept «Time»

Anticorruption Layer

`Get{time}`



- Reference: DCF77
- Resolution: 1 sec
- Operation: continuous

- Reference: quartz impulses
- Resolution: 1 msec
- Operation: counter
- Nulled at power-up

Example:
Concept «Time»

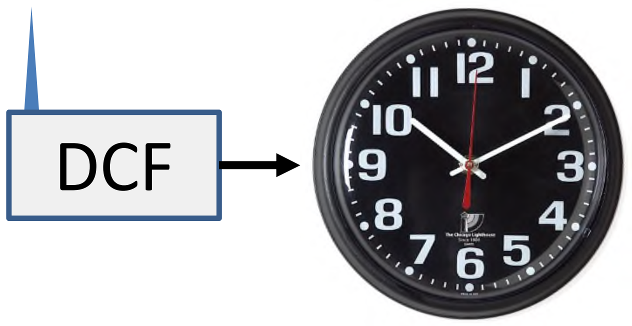
Anticorruption Layer

Agree on:
- Common (external) reference
- Exchange format

22:09:01

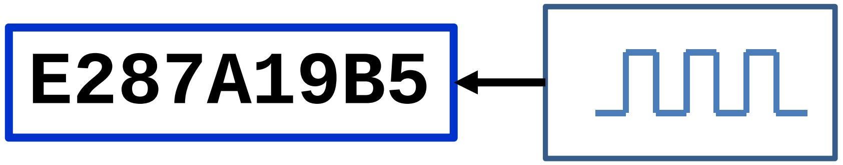
E287A19B5

System «A»



- Reference: DCF77
- Resolution: 1 sec
- Operation: continuous

System «B»



- Reference: quartz impulses
- Resolution: 1 msec
- Operation: counter
- Nulled at power-up

https://www.pinterest.com

Summary: The DSE concepts:

Business/Application Domain — Sphere of Knowledge, Influence or Activity

Bounded Context — The Bounded Context is the explicit Boundary of a Model

Domain Model — "Formal" representation of the Entities, Relationships and their Properties in your Domain

Anticorruption Layer — Maintains the integrity of differing systems and models

DSE

Fundamentals (2/2)

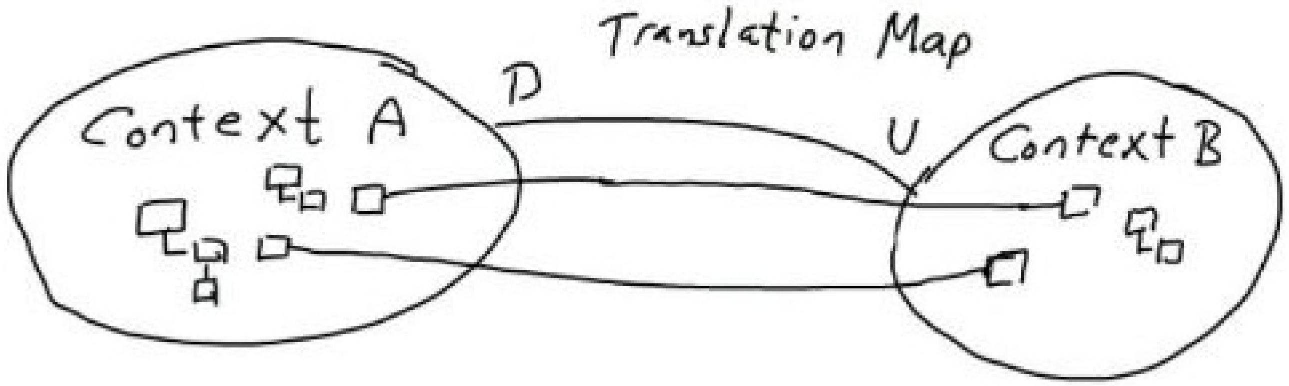
The Tools:

Context Map

Ubiquitous Language

Domain-Specific Language

DSE Patterns



http://aidium.se

http://www.iconshut.com

Context Map =

A Context Map is a document which defines and delineates the different bounded contexts for systems/models and the relationships between them

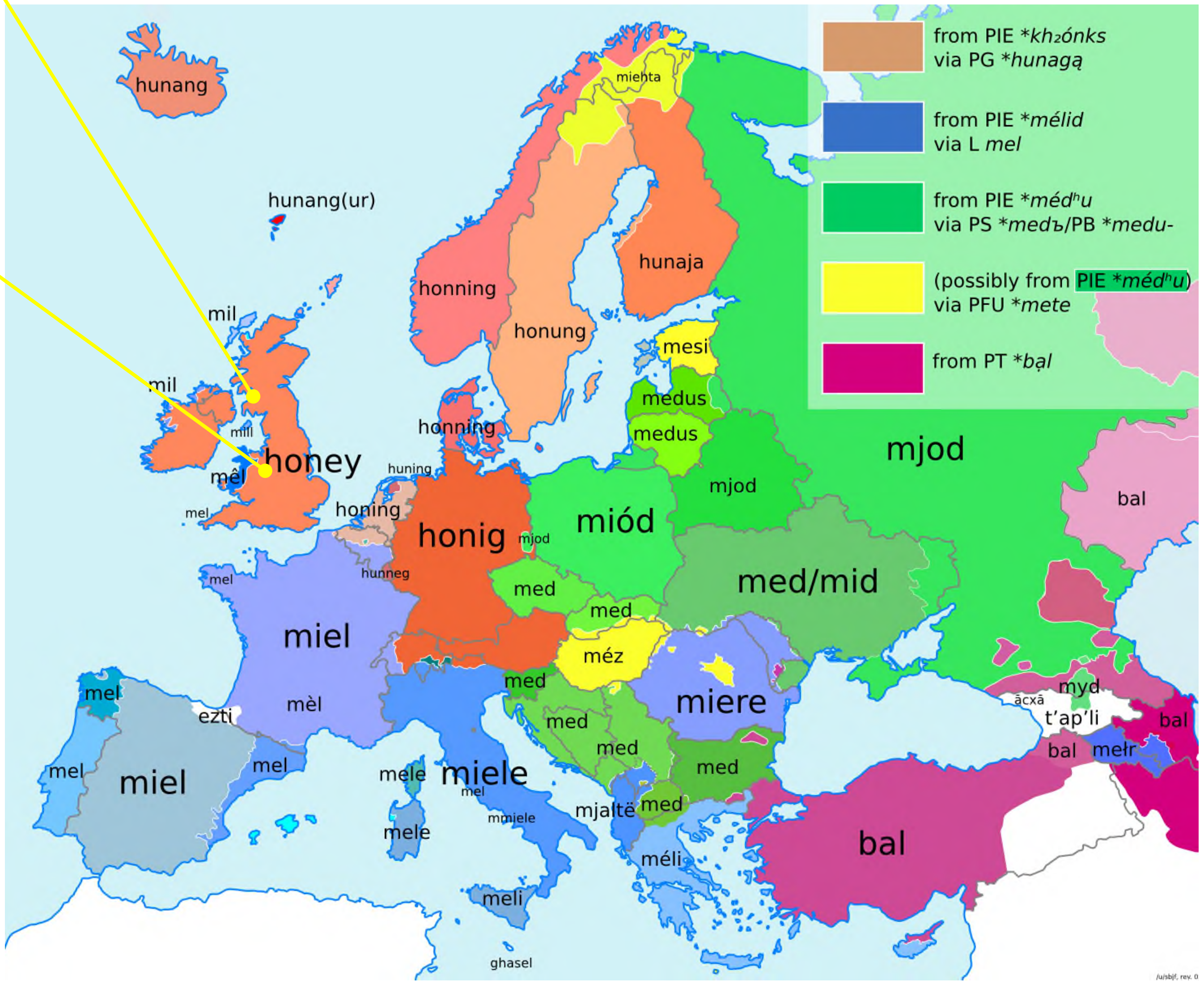
Example: Context = Country

<u>Bounded Context </u>= Country

<u>Concepts</u> =
- Language
  - Laws
    - …

"Pool" =
Schwimmbecken
Fonds
Poolbillard
Bassin
Tümpel
Einsatz
Fundus
gemeinsame Spielkasse
Grube
Interessengemeinschaft
Konsortium
Reservoir
Ring

http://dict.leo.org



http://indo-europeanlanguages.blogspot.ch

Context Map

http://rotrwarzone.boards.net

http://www.crainscleveland.com

http://securities.clarksons.com

Can we model the whole world in one model?

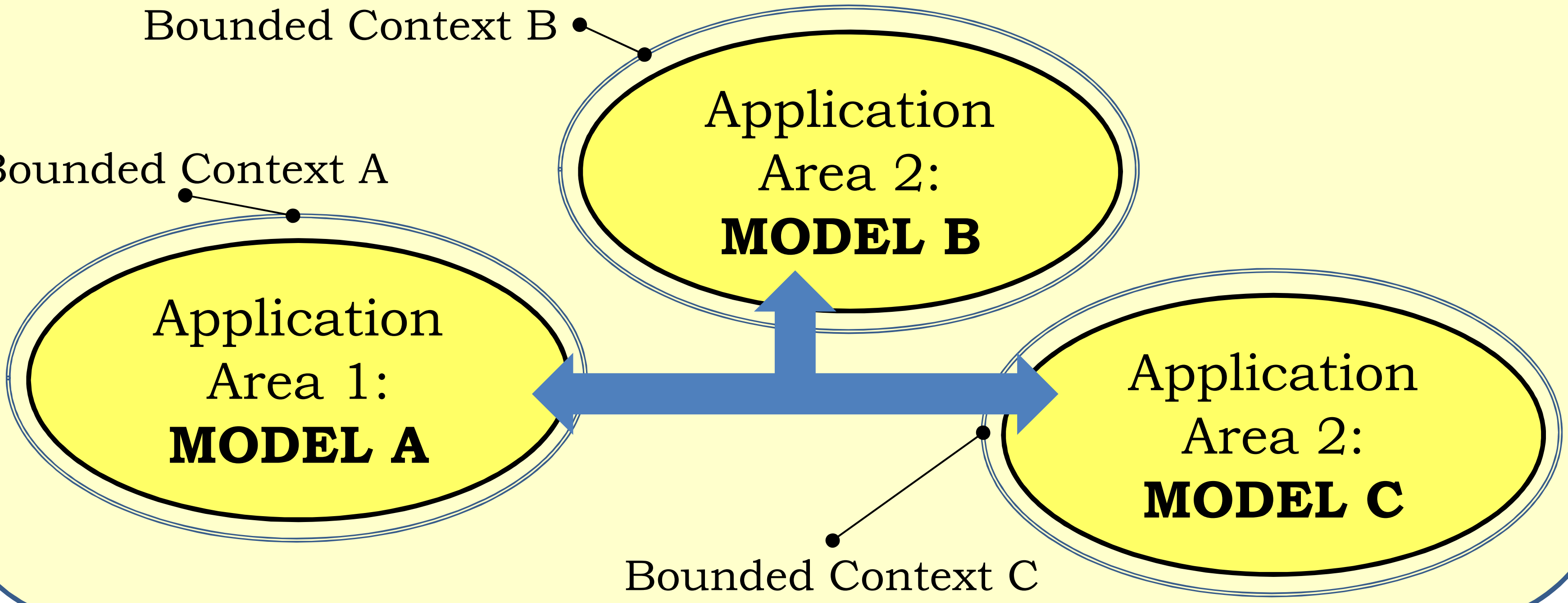
A complete Enterprise?

A department?

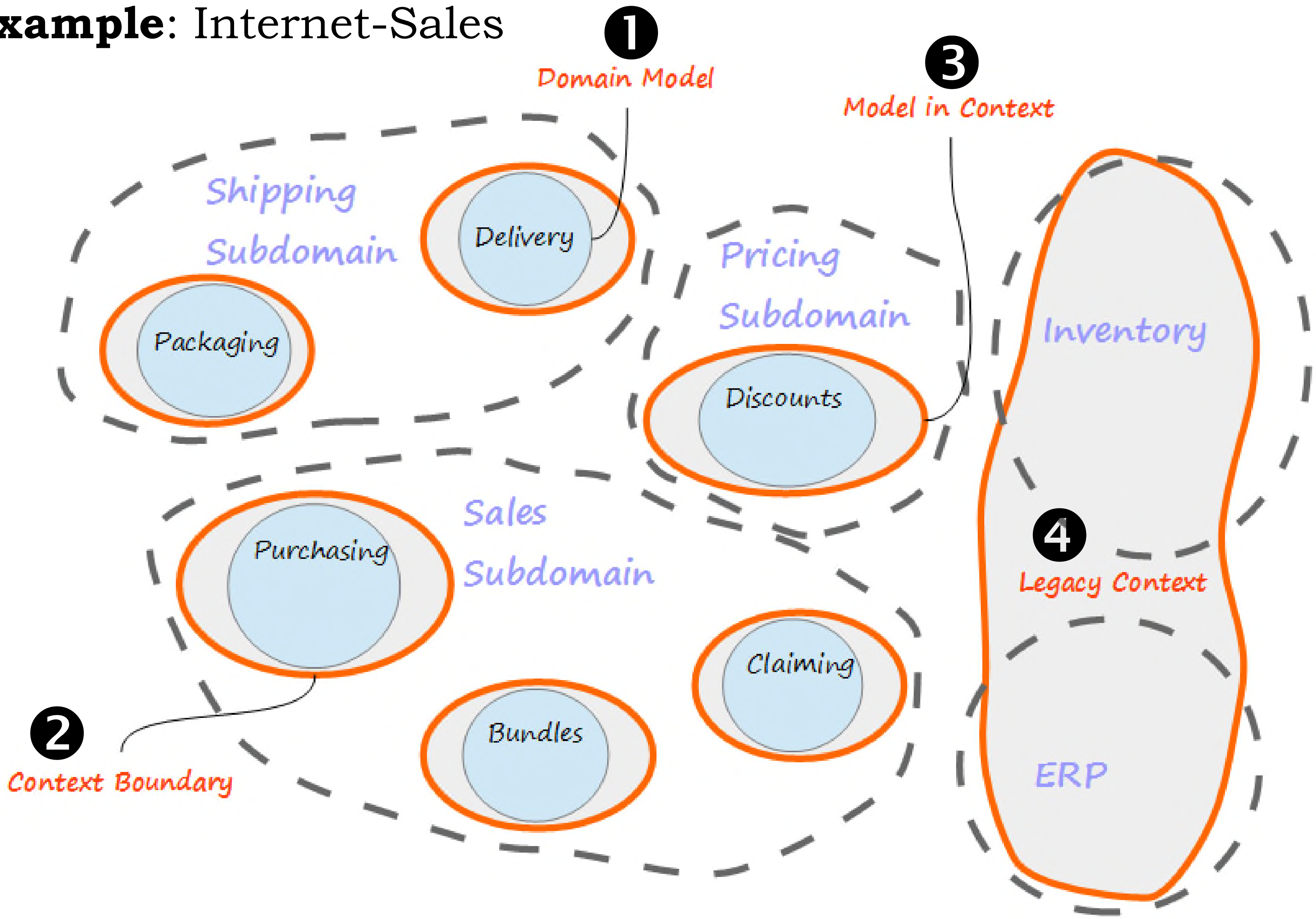
NO – We need clearly defined *boundaries* for our models http://www.crainscleveland.com

Context Map

Context Map =
Definition of different bounded contexts and the relationships between them

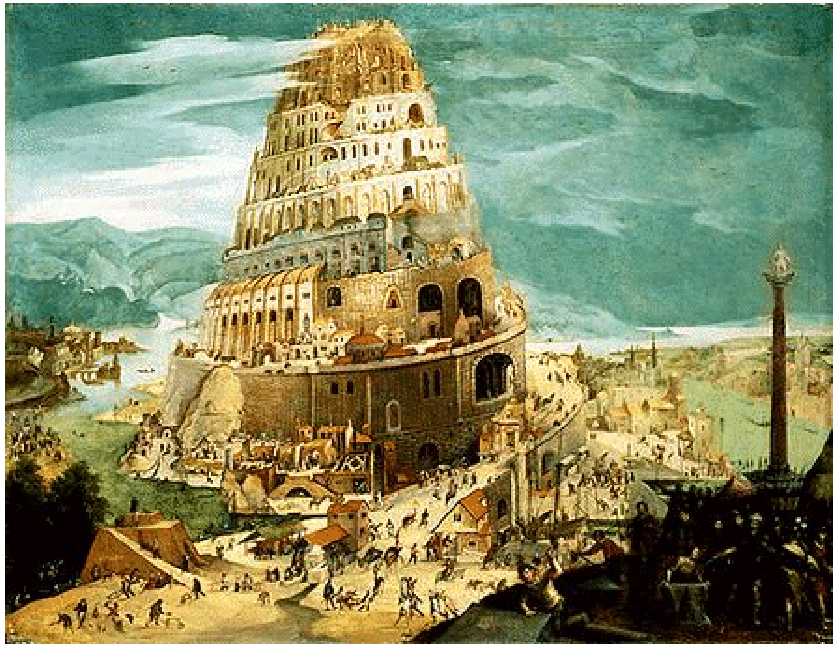

Bounded Context B
Application Area 2: MODEL B
Bounded Context A
Application Area 1: MODEL A
Application Area 2: MODEL C
Bounded Context C

Example: Internet-Sales



https://blog.codecentric.de

http://joseph_berrigan.tripod.com



Ubiquitious Language

A major reason for failure of software projects is a failure of **people** = the failure to *communicate*

DEFINITIONS

Ubiquitious Language [UL] =

The Ubiquitous Language is a *shared* language between the business and the development teams

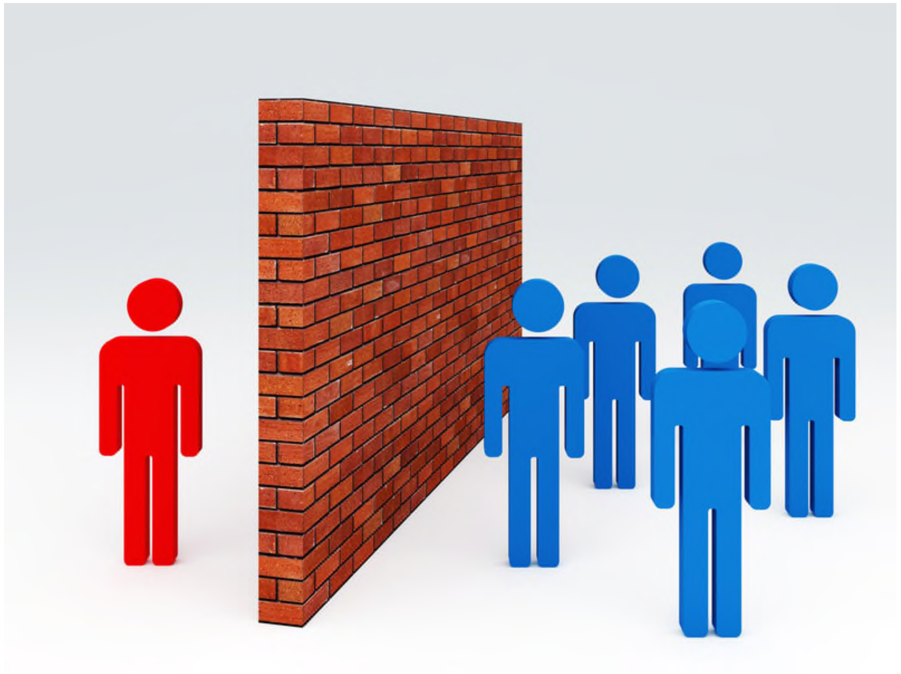
The Ubiquitous Language comes from the *business*, and is enriched by the development team

www.thinkddd.com

Ubiquitious Language

http://mayrsom.com

Business Customer
Needs, Requirements



Information Systems Engineers
Specifications, Implementation

- Business *vocabulary*
- *Implicit* knowledge
- No *model*

- IT *vocabulary*
- *Implicit* knowledge
- IT *models*

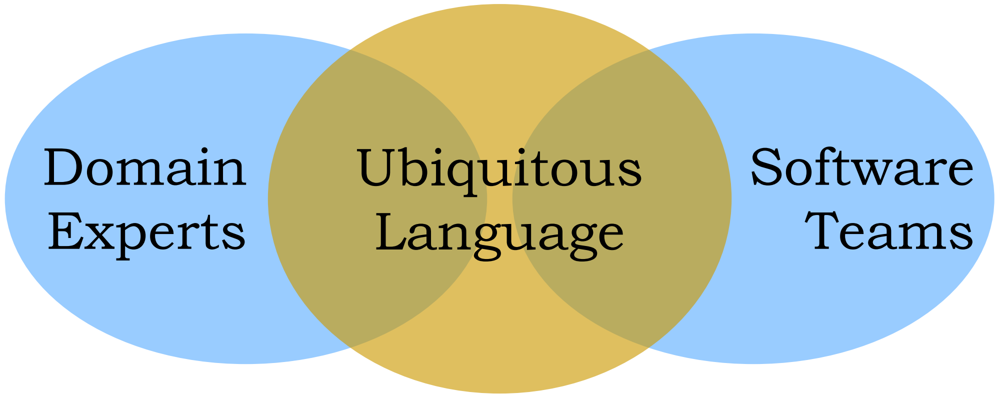
serious communications gap

Customer/Business

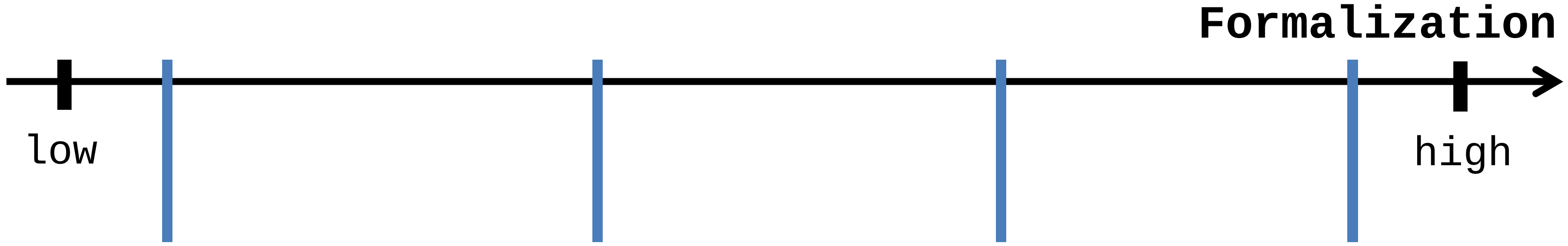
UL

IT Organization

http://clipartzebraz.com



https://cimx.wordpress.com



«Boxes & Lines» Text | Boxes & Lines with semantics | UML, SysML | Ontologies

How is an Ubiquitous Language developed?

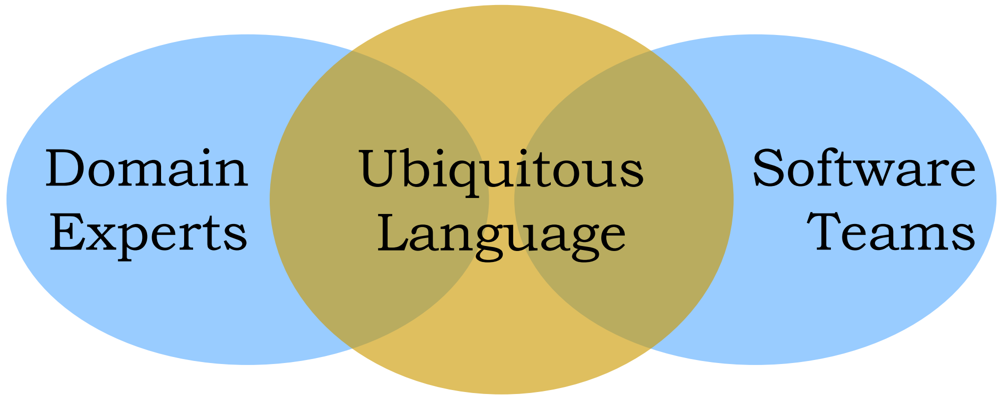
… very often a good **start** is a textual table

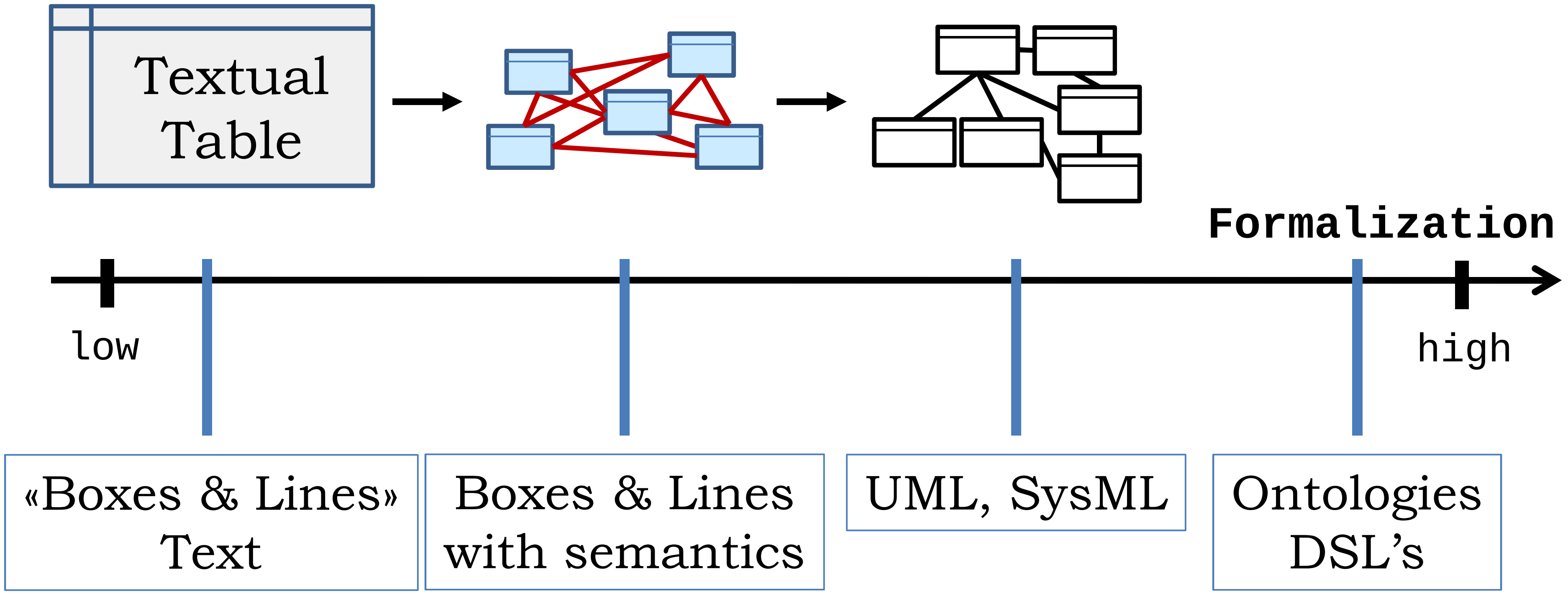
High Level Domain Entities (Enterprise Level)		
Domain Concept	**Description**	**Operations**
Organization Entity	Legal Entity for executing business	• Create the entity • Internal organization of the entity • Agreements with other parties • Creation of financial products • Collaborate with other parties • Create reports • …
Operation	Value-transferring activity with adherence to legal & regulatory requirements	• Define parties • Oblige parties • Check legal & regulatory requirements • Execute operation • Document & archive operation • …
etc.		
etc.		

Definition

Concepts

Operations


Domain Experts
Ubiquitous Language
Software Teams

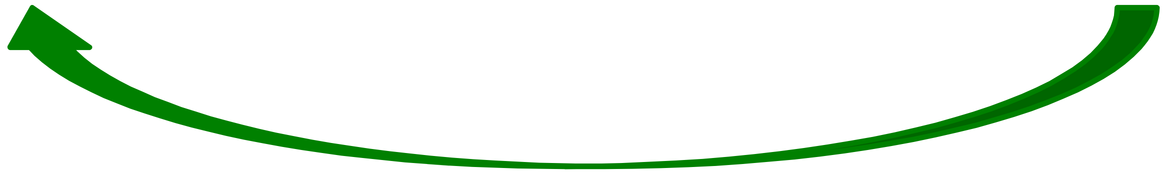

Textual Table
Formalization
low
high
«Boxes & Lines» Text
Boxes & Lines with semantics
UML, SysML
Ontologies DSL's

How is an Ubiquitous Language developed?

Domain Expert

IT Expert

Content: Concepts, Behaviour, Constraints



Form: Syntax, Enrichment, Precision

http://blog.asha.org

http://www.iconshut.com

Domain Specific Language =

A computer programming language of limited expressiveness focused on a particular domain.

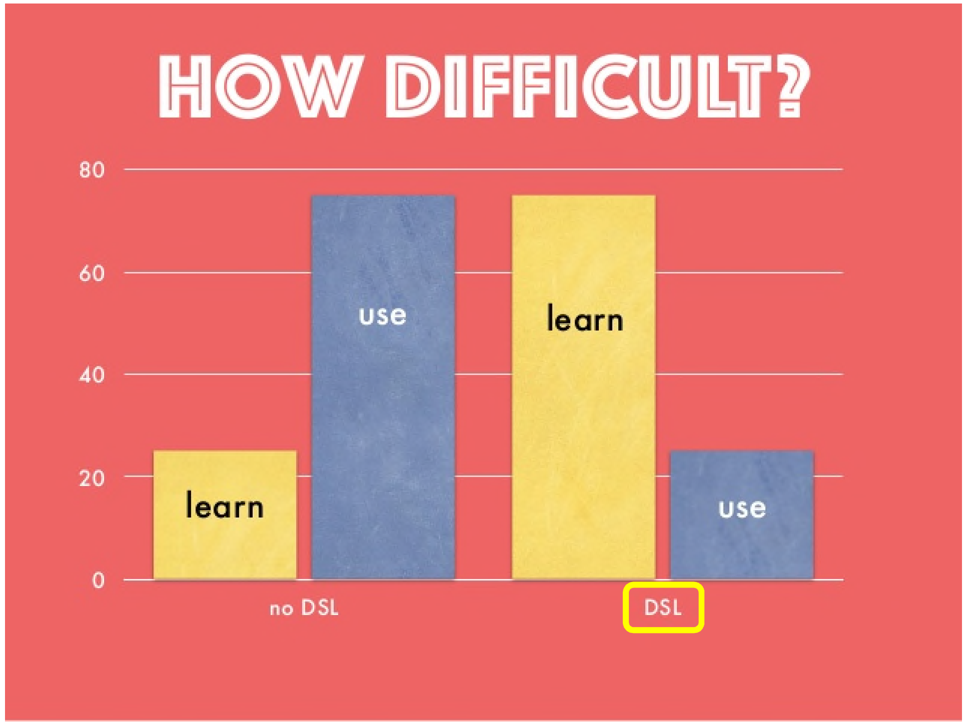
The domain focus is what makes a limited language worthwhile.

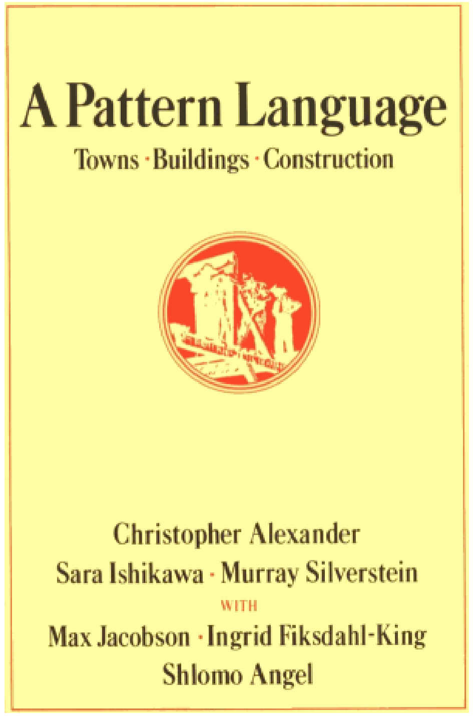
Fowler/Parsons 2011

Domain-specific languages (DSLs) are currently being developed for many application domans, e.g. insurance, banking, robotics, …

⇒ They *may* prove useful in specific fields of application (Risk: Dilution)

http://de.slideshare.net/netguru/sparkcamp-jakubnaliwajek




A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel

http://www.urbagram.net

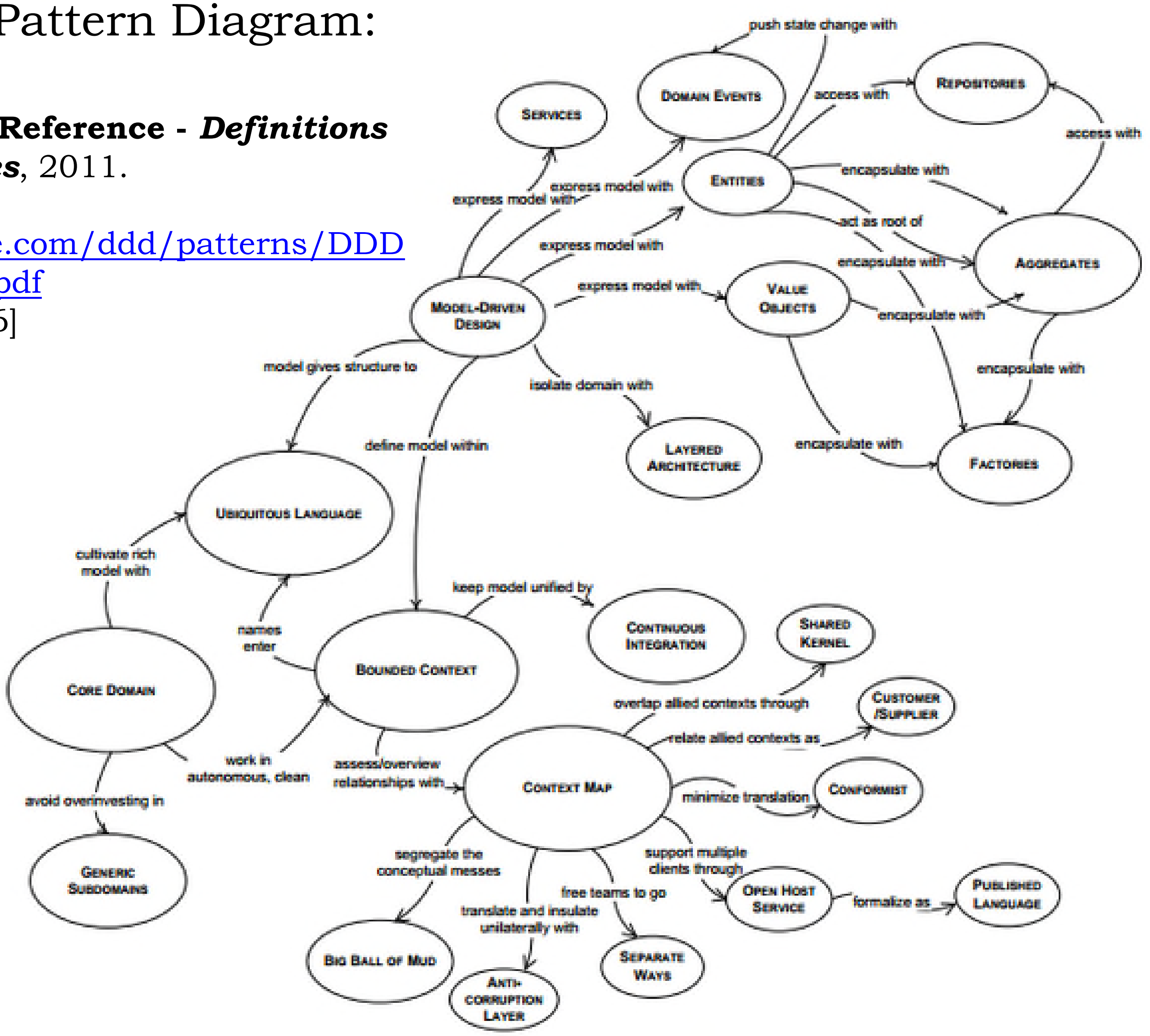
http://www.iconshut.com

DSE Patterns =

Tried and true design and development reference solutions for Domain-Driven Design (& Domain Software Engineering)

Eric Evans DDD Pattern Diagram:

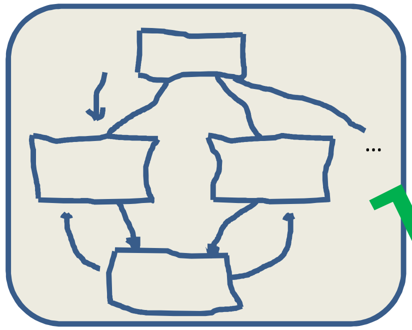
Eric Evans:
Domain-Driven Design Reference - *Definitions and Pattern Summaries*, 2011.
Downloadable from:
https://domainlanguage.com/ddd/patterns/DDD_Reference_2011-01-31.pdf
[last accessed: 29.1.2016]

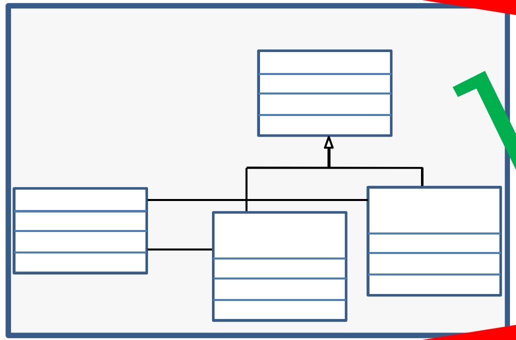


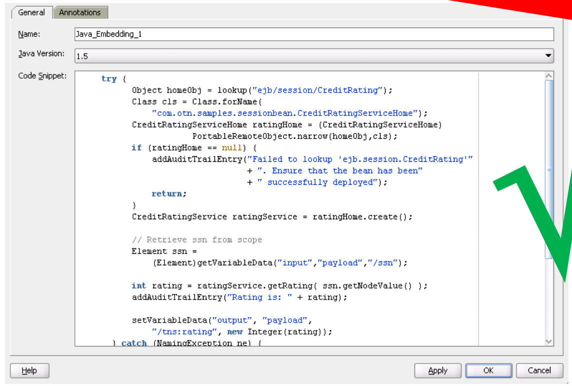
DSE

Alignment & Continuous Integration

Loss of Consistency: Continuous Alignment







Name: Java_Embedding_1
Java Version: 1.5
Code Snippet:

```
    try {
        Object homeObj = lookup("ejb/session/CreditRating");
        Class cls = Class.forName(
            "com.otn.samples.sessionbean.CreditRatingServiceHome");
        CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
                    PortableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                               + ". Ensure that the bean has been"
                               + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

Time, Evolution

Loss of Consistency

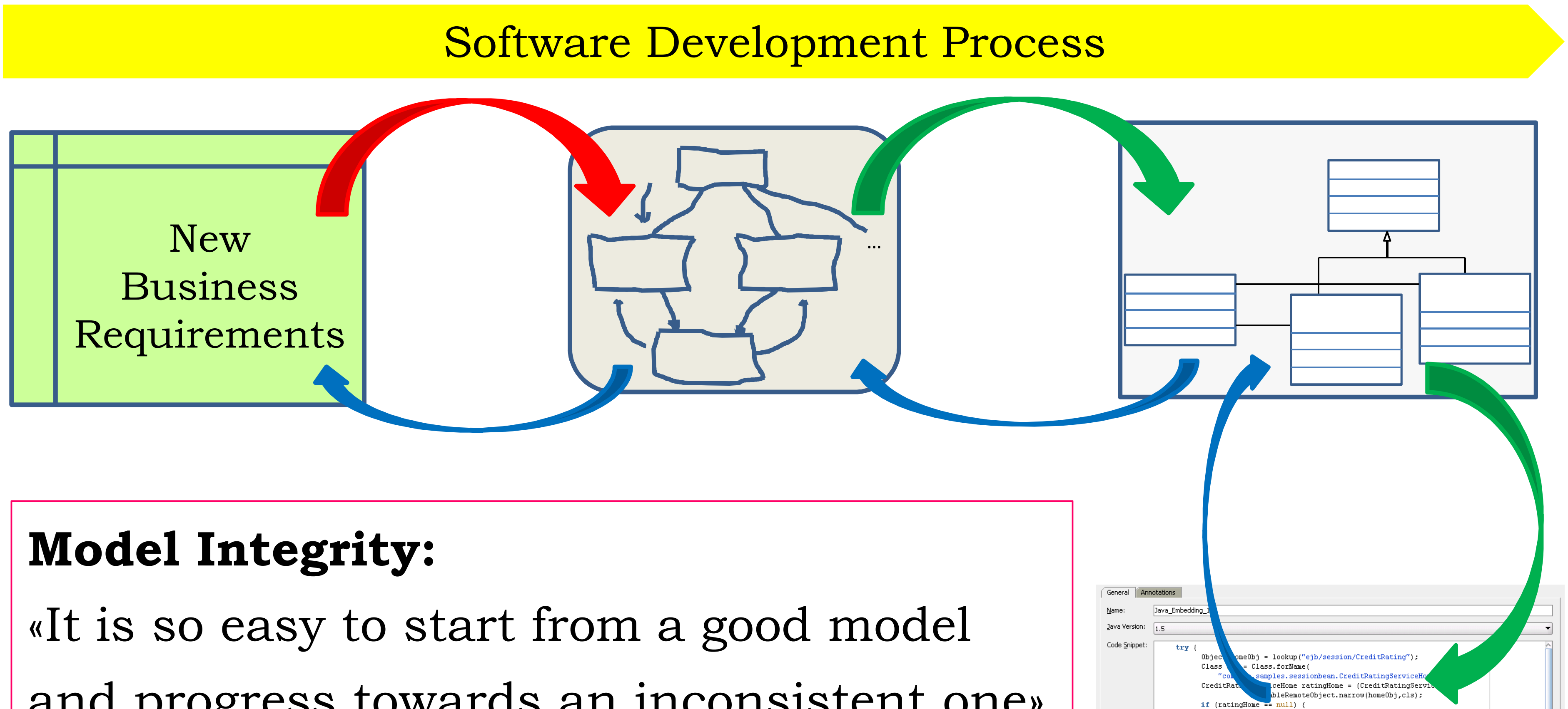
Loss of Consistency

Business Model ⇔
Domain Model ⇔
Code

must be in sync at all times

http://publicdomainvectors.org

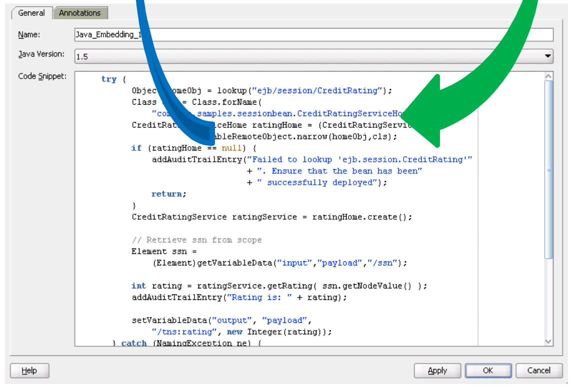
Continuous Alignment



Model Integrity:

«It is so easy to start from a good model and progress towards an inconsistent one»

… *and the correctness of the software becomes suspect and its agility is damaged*



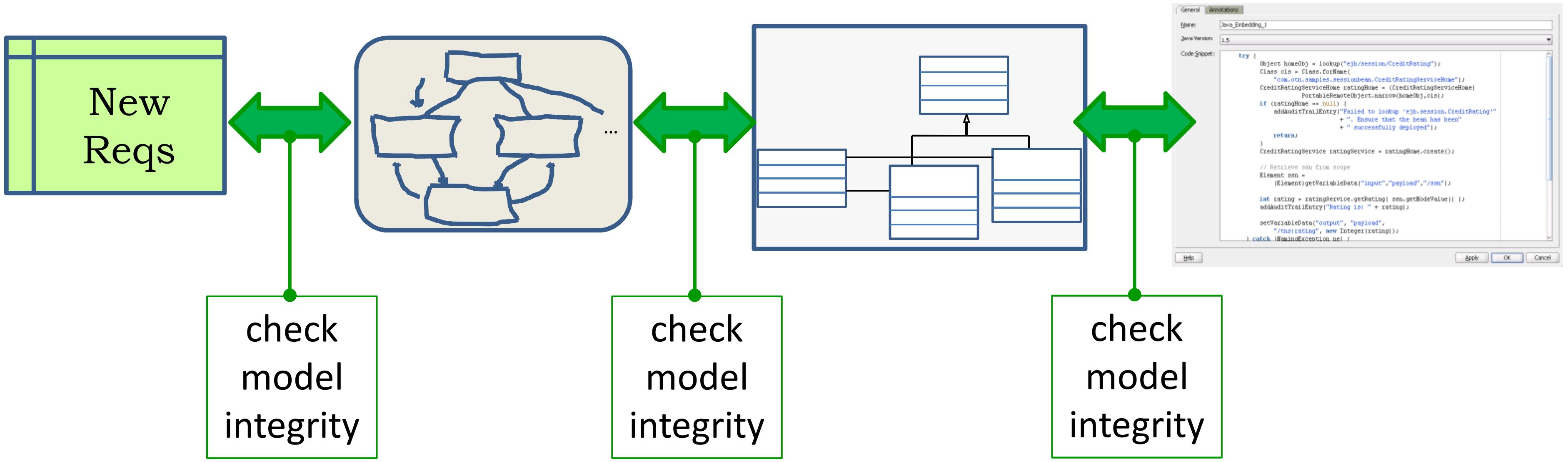
Name: Java_Embedding_1

Java Version: 1.5

Code Snippet:

```
    try {
        Objec[illegible]omeObj = lookup("ejb/session/CreditRating");
        Class [illegible] = Class.forName(
            "co[illegible]samples.sessionbean.CreditRatingServiceHo[illegible]
        CreditRat[illegible]iceHome ratingHome = (CreditRatingServic[illegible]
                  [illegible]ableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                               + ". Ensure that the bean has been"
                               + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

Continuous Integration



Name: Java_Embedding_1

Java Version: 1.5

```
    try {
        Object homeObj = lookup("ejb/session/CreditRating");
        Class cls = Class.forName(
            "com.otn.samples.sessionbean.CreditRatingServiceHome");
        CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
                    PortableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                              + ". Ensure that the bean has been"
                              + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

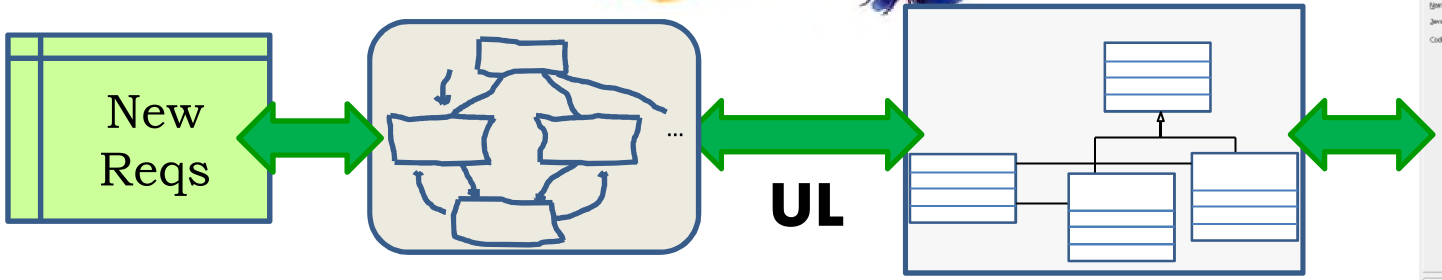
«It is easy to make mistakes when we do not focus 100% on the purity, integrity and consistency of the model»

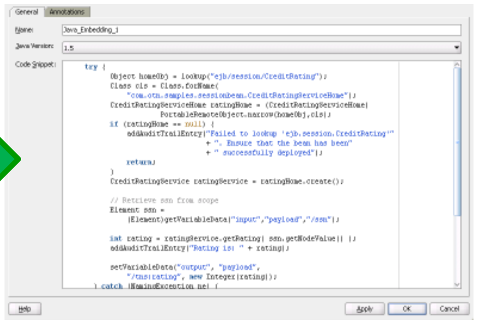
Continuous integration is based on integration of concepts in the model, then finding its way into the implementation where it is tested

DSE

Consequences for Industrial Software Development

Domain Software Engineering (DSE)
is an *architectural methodology* for evolving a software system that closely aligns to *business domains*





Name: Java_Embedding_1
Java Version: 1.5
Code Snippet:

```
    try {
        Object homeObj = lookup("ejb/session/CreditRating");
        Class cls = Class.forName(
            "com.otn.samples.sessionbean.CreditRatingServiceHome");
        CreditRatingServiceHome ratingHome = (CreditRatingServiceHome)
                    PortableRemoteObject.narrow(homeObj,cls);
        if (ratingHome == null) {
            addAuditTrailEntry("Failed to lookup 'ejb.session.CreditRating'"
                                 + ". Ensure that the bean has been"
                                 + " successfully deployed");
            return;
        }
        CreditRatingService ratingService = ratingHome.create();

        // Retrieve ssn from scope
        Element ssn =
            (Element)getVariableData("input","payload","/ssn");

        int rating = ratingService.getRating( ssn.getNodeValue() );
        addAuditTrailEntry("Rating is: " + rating);

        setVariableData("output", "payload",
            "/tns:rating", new Integer(rating));
    } catch (NamingException ne) {
```

- Massively business-oriented
- Strongly model-based/model-driven
- Continuous integration

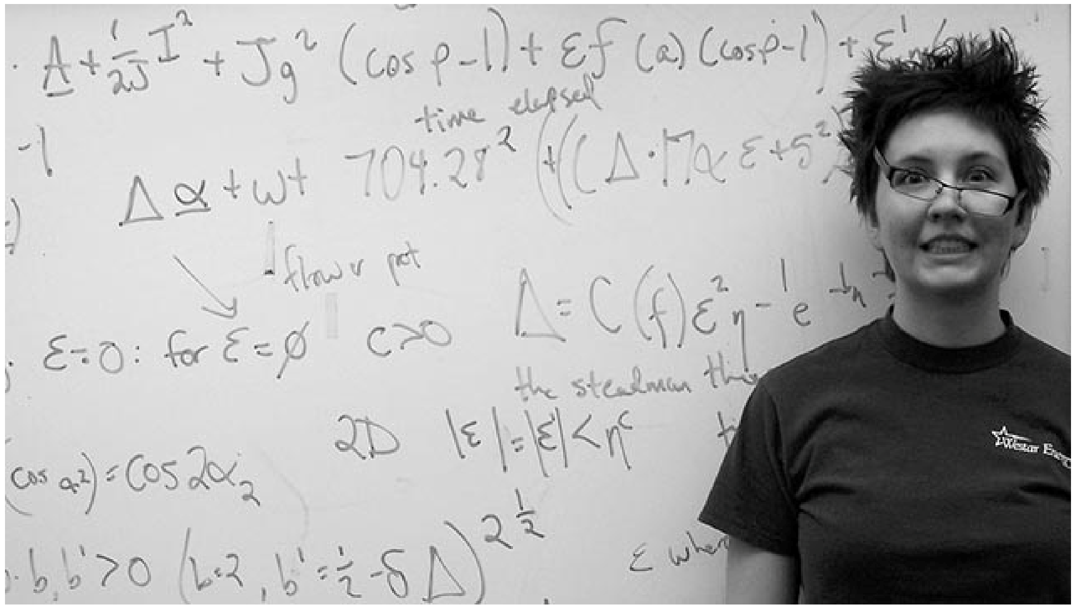
DSE Expected Benefits:

- ✓ Precise requirements
- ✓ Minimal divergence business reqs ⇔ IT implementation
- ✓ Reduction of accidental complexity
- ✓ Significant increase in software quality
- ✓ Optimum agility/maintainability of the software
- ✓ Excellent understandability

http://www.toyourhealth.com

DSE Obstacles:

- ❖ Strict discipline needed
- ❖ Willing business partners
- ❖ Adequate development process & governance
- ❖ Very competent people
- ❖ Continuous refactoring
- ❖ Up-front investments (for each project)



http://lifehacker.com

Is DSE difficult?

YES

http://www.fbnportal.com

Is DSE worthwhile?

YES – but needs a strong committment and much company discipline

DSE

Conclusions

What have we learned?

http://instructionaldesign.org

- ✓ The Software Domain Engineering Concepts
- ✓ The Software Domain Engineering Tools
- ✓ The Pro's and Con's of DSE

Why is this knowledge good for you?

- ✓ DSE is becoming an important methodology in the (near?) future
- ✓ DSE has a number of wonderful concepts
- ✓ Introduction of DSE needs time/effort

DSE

References

References (1/3):

References	
Evans04	Eric Evans: **Domain-Driven Design – *Tackling Complexity in the Heart of Software*** Addison-Wesley, Boston, USA, 2004. ISBN 0-321-12521-5
Duffy04	Daniel Duffy: **Domain Architectures – *Models and Architectures for UML Applications*** John Wiley & Sons, Inc., Chichester, UK, 2004. ISBN 0-470-84833-2
Nilsson06	Jimmy Nilsson: **Applying Domain-Driven Design and Patterns – *with Examples in C# and .NET*** Addison-Wesley (Pearson Education), NJ, USA, 2006. ISBN 978-0-321-26820-4
Vernon13	Vaughn Vernon: **Implementing Domain-Driven Design** Addison-Wesley (Pearson Education), NJ, USA, 2013. ISBN 978-0-321-83457-7
Bjørner06	Dines Bjørner: **Software Engineering 3 – *Domains, Requirements, and Software Design*** Springer-Verlag, Berlin DE, 2006. ISBN 978-3-540-21151-8
Kleppe09	Anneke Kleppe: **Software Language Engineering – *Creating Domain-Specific Languages Using Metamodels*** Addison-Wesley (Pearson Education), NJ, USA, 2009. ISBN 978-0-321-55345-4
Voelter13	Markus Voelter: **DSL Engineering – *Designing, Implementing and Using Domain-Specific Languages*** Dslbook.org, 2013. ISBN 978-1-48121-858-0

References (2/3):

References	
Millett15	Scott Millett, Nick Tune: **Patterns, Principles, and Practices of Domain-Driven Design** John Wiley & Sons, Inc., Indianapolis, USA, 2015. ISBN 978-1-118-71470-6
Reinhartz13	Iris Reinhartz-Berger, Arnon Sturm, Tony Clark, Sholom Cohen, Jorn Bettin (Editors): **Domain Engineering – *Product Lines, Languages, and Models*** Springer-Verlag, Berlin, 2013. ISBN 978-3-642-36653-6
Scotts14	Jason Scotts: **Domain-Driven Desing – *How to easily implement Domain-Driven Design*** Printed in Germany by Amazon Distribution, Leipzig, 2014. ISBN 978-1-631-87691-2
Evans15	Eric Evans: **Domain-Driven Design Reference – *Definitions and Pattern Summaries*** Dog Ear Publishing,Indianapolis, USA, 2015. ISBN 978-1-45750-119-7
Avram06	Abel Avram, Floyd Marinescu: **Domain-Driven Design - *Quickly*** C4Media Inc., USA, 2006. ISBN 978-1-4116-0925-9 Download: http://www.infoq.com/minibooks/domain-driven-design-quickly
Kelly08	Steven Kelly, Juha-Pekka Tolvanen: **Domain-Specific Modeling – *Enabling Full Code Generation*** John Wiley & Sons, Inc., Hoboken, N.J., USA, 2008. ISBN 978-0-470-03666-2
Gonzales08	Cesar Gonzales-Perez, Brian Henderson-Sellers: **Metamodelling for Software Engineering** John Wiley & Sons., Chichester, UK, 2008. ISBN 978-0-470-03036-3

References (3/3):

References	
Fowler11	Martin Fowler: **Domain-Specific Languages** Addison-Wesley (Pearson Education), NJ, USA, 2011. ISBN 978-0-321-71294-3
OSE03	Operating System Engineering: **Domain Engineering** 2003, Downloadabel from: https://www4.cs.fau.de/Lehre/SS03/V_OSE/Skript/03ose-A5.pdf [last accessed: 2.12.1015]
Henderson12	Brian Henderson-Sellers: **On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages** Springer-Verlag, Heidelberg, 2012. ISBN 978-3-642-29824-0
Rumbaugh05	James Rumbaugh, Ivar Jacobson, Grady Booch: **The Unified Modeling Language UML Reference Manual** Addison-Wesley (Pearson Education), NJ, USA, 2005. ISBN 978-0-321-24562-8
Sølvberg10	Arne Sølvberg: **Domain Engineering: What is it?** I. Reinhartz-Berger, A. Sturm, Y. Wand, J. Bettin, T. Clark, S. Cohen, J. Ralyté, and P. Plebani (Eds.): CAiSE 2010 Workshop DE@CAiSE'10, Hammamet, Tunisia, pp. 1-5, 2010. Downloadable from: http://ceur-ws.org/Vol-602/DE_CAiSE10_paper1_Solvberg.pdf [last accessed: 2.12.1015]
Bjørner05	Dines Bjørner: **Domain Engineering** Downloadable from: http://www.imm.dtu.dk/~dibj/facs-domain.pdf [last accessed: 2.12.1015]
Taylor10	Richard N. Taylor, Nenad Medvidovic, Eric M. Dashofy: **Intro to Domain-Specific Software Engineering** Downloadabel from: http://sunset.usc.edu/classes/cs578_2009b/23_Intro_to_DSSE.ppt

Questions please