OOCASE V4.0.4 Overview

Tekn. Dr. Olof Johansson
ROJTEC
Tel: 070-6577608
olojo@rojtec.se

Contents

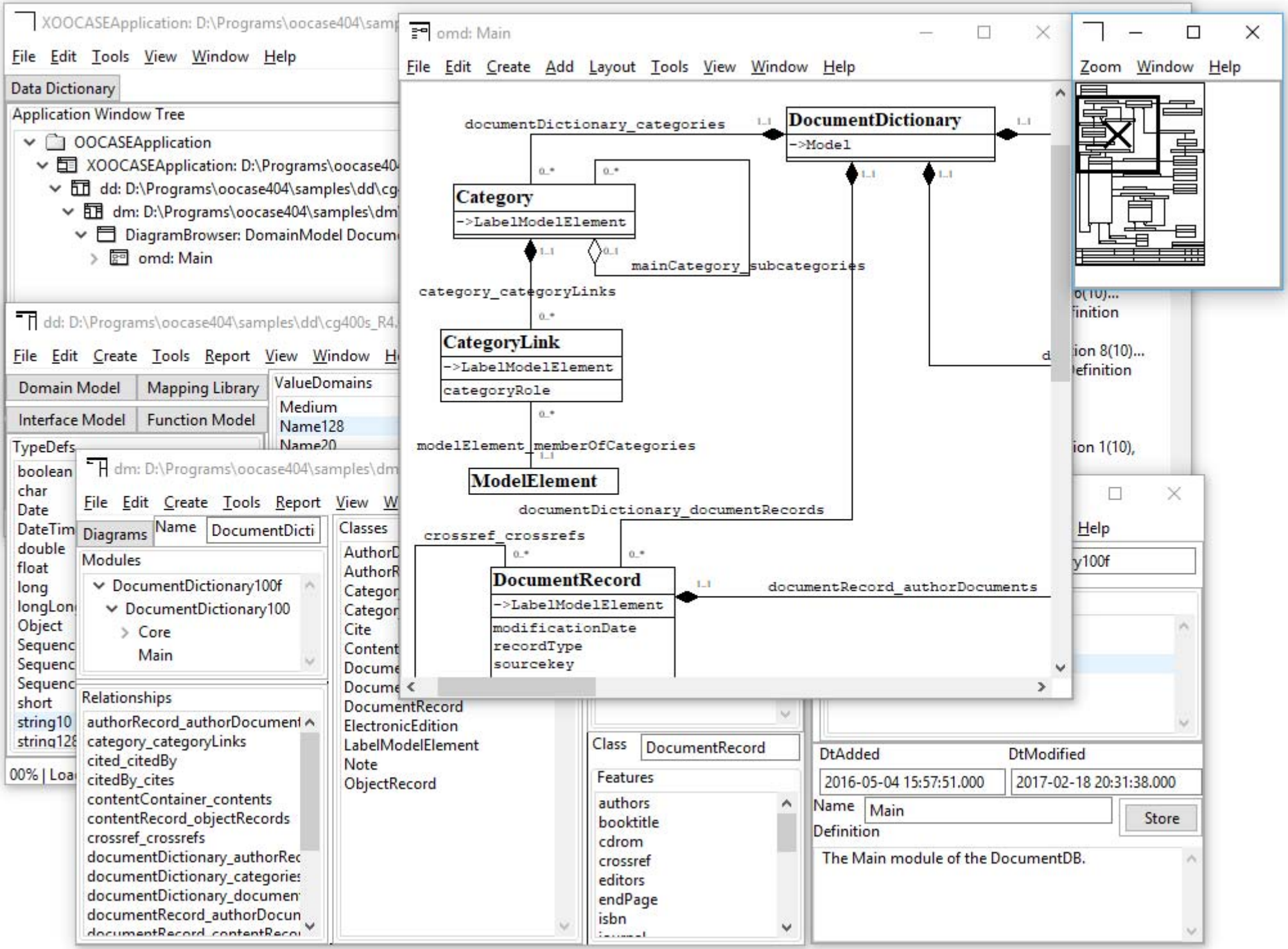
1 Object-Oriented Computer Aided Software Engineering (OOCASE)

Design Software with Computer Aided Design Tools

- An Object-Oriented DomainModel specifies Classes, Inheritance, Attributes and Relationships for a Software Application
  - - the DomainModel is stored as a .odm file or in a relational database
- Reuse Modules with Standard Classes, Attribues and Relationships
- Copy/Paste Classes, Attributes and Relationships
- Design graphically with ObjectModelDiagrams (UML Class diagrams)
- Use a creative interactive design environment for software applications that contain hundreds of classes, relationships and attributes

DataDictionary

- Store Object-Oriented DomainModels in a relational database repository. Generate source code automatically from the models.
- Use the repository to search for, load, store, archive, and collaborate on versions and releases of Object-Oriented DomainModels.


XOOCASEApplication: D:\Programs\oocase404\samp
File Edit Tools View Window Help
Data Dictionary
Application Window Tree
OOCASEApplication
XOOCASEApplication: D:\Programs\oocase40
dd: D:\Programs\oocase404\samples\dd\cg
dm: D:\Programs\oocase404\samples\dm
DiagramBrowser: DomainModel Docum
omd: Main
omd: Main
File Edit Create Add Layout Tools View Window Help
documentDictionary_categories
DocumentDictionary
->Model
Category
->LabelModelElement
mainCategory_subcategories
category_categoryLinks
CategoryLink
->LabelModelElement
categoryRole
modelElement_memberOfCategories
ModelElement
documentDictionary_documentRecords
crossref_crossrefs
DocumentRecord
->LabelModelElement
modificationDate
recordType
sourcekey
documentRecord_authorDocuments
Zoom Window Help
dd: D:\Programs\oocase404\samples\dd\cg400s_R4
File Edit Create Tools Report View Window H
Domain Model
Mapping Library
Interface Model
Function Model
ValueDomains
Medium
Name128
Name20
TypeDefs
boolean
char
Date
DateTim
double
float
long
longLon
Object
Sequenc
Sequenc
Sequenc
short
string10
string128
dm: D:\Programs\oocase404\samples\dm
File Edit Create Tools Report View W
Diagrams
Name
DocumentDicti
Modules
DocumentDictionary100f
DocumentDictionary100
Core
Main
Relationships
authorRecord_authorDocument
category_categoryLinks
cited_citedBy
citedBy_cites
contentContainer_contents
contentRecord_objectRecords
crossref_crossrefs
documentDictionary_authorRec
documentDictionary_categories
documentDictionary_documen
documentRecord_authorDocun
Classes
AuthorD
AuthorR
Categor
Categor
Cite
Content
Docume
Docume
DocumentRecord
ElectronicEdition
LabelModelElement
Note
ObjectRecord
Class
DocumentRecord
Features
authors
booktitle
cdrom
crossref
editors
endPage
isbn
Help
y100f
DtAdded
DtModified
2016-05-04 15:57:51.000
2017-02-18 20:31:38.000
Name
Main
Store
Definition
The Main module of the DocumentDB.

2 Quality Assurance with Edition, Version and Release (QAEVR)

Quality Assurance - after creative interactive design has outlined the application DomainModel

- Run thousands of automated checks on a DomainModel
- Interactively fix errors and warnings at once in the DomainModel
- Stamp ModelElements as CheckedBy, and dtChecked datetimestamp
- Stamp ModelElements as ApprovedBy, and dtApproved

Edition E{year}-{month}-{day} {hour}:{minute}:{second} E2018-12-13 16:21:00

- Edited copies of previously quality assured ModelElements from reused DomainModels, automatically become Editions with full traceability to the version and release of the original ModelElement.

Version V{major}.{minor}.{patch} V1.0.1

After Quality Assurance of an edited/redesigned DomainModel ;

- Versions are assigned to each individual ModelElement according to SemVer 2.0.0. V{major}.{minor}.{patch}.{optionar suffixes}
- All edited ModelElements in the whole DomainModel can be assigned the same Version with one command.

Release R{major}.{minor}.{patch} R1.0.1

- A release is on ONE DomainModel as a WHOLE, and written on all objects in the model

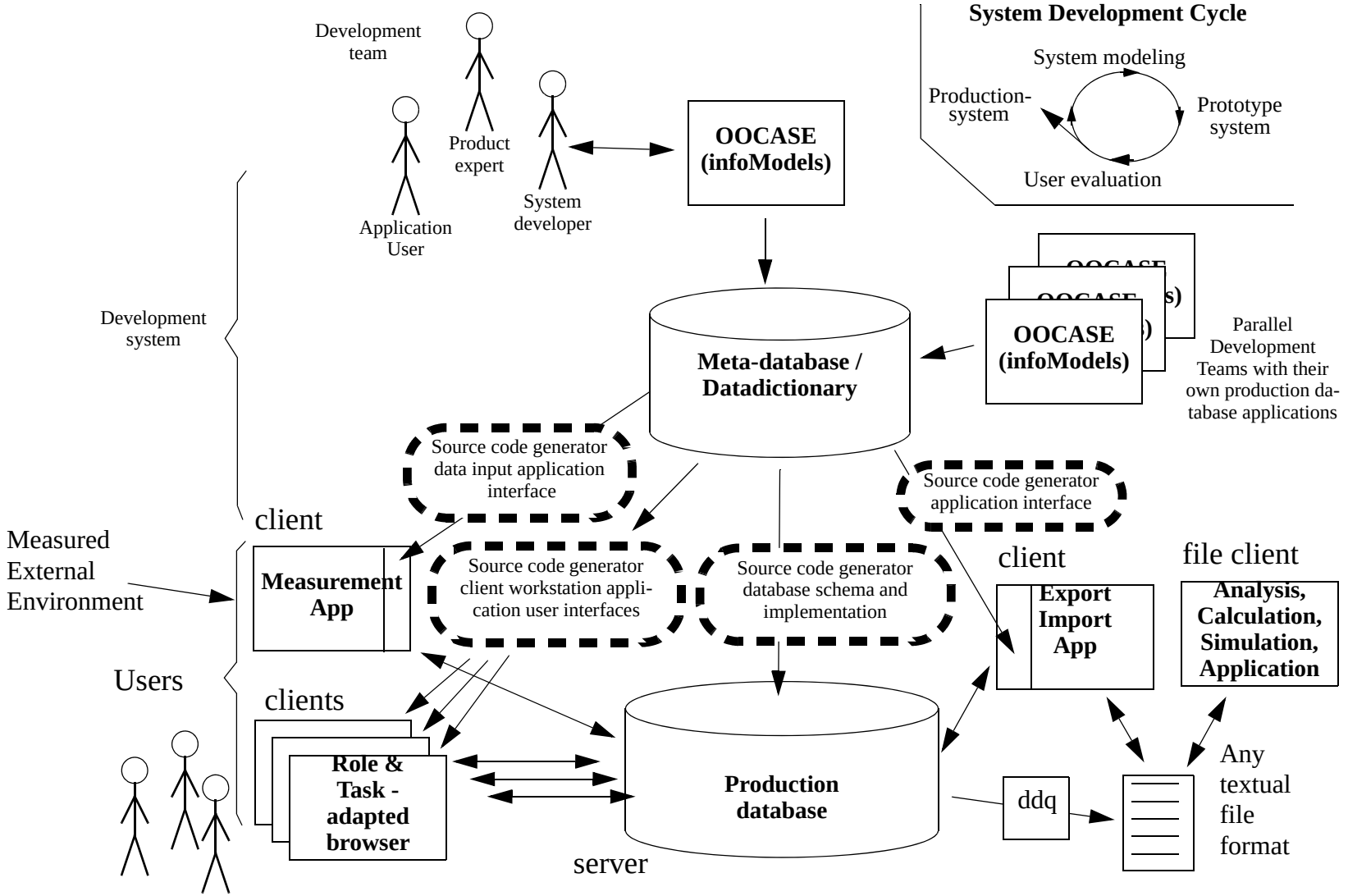
3 DomainModel driven Source Code Generation 1(3)

Source Code Generation of Application Programming Interfaces (APIs)

- The DomainModel designed in OOCASE defines the applications' Classes, Inheritance, Attributes and Relationships
- The DomainModel defines the applications' object information Structure AND from the Structure directly inferable object Behaviour.
- Inferable behavior that implements from the structure inferable application programming interfaces (API's) that follow easy to remember and use naming and parameter setting conventions.
- The APIs are implemented by configurable automated source code generators specialized for a highly optimized implementation in a particular target source code programming language.
- Complete Software Module Libraries with Standard Object Manipulation Functionality on objects, their attribute values and relationships are generated from the DomainModel with parameterizied sourcecode templates

A Source Code Generator library is a ONE-TIME quality assurance investment that can be reused for generating any number of implementations of any DomainModel. We have used ours for decades.

3 DomainModel driven Source Code Generation 2(3)

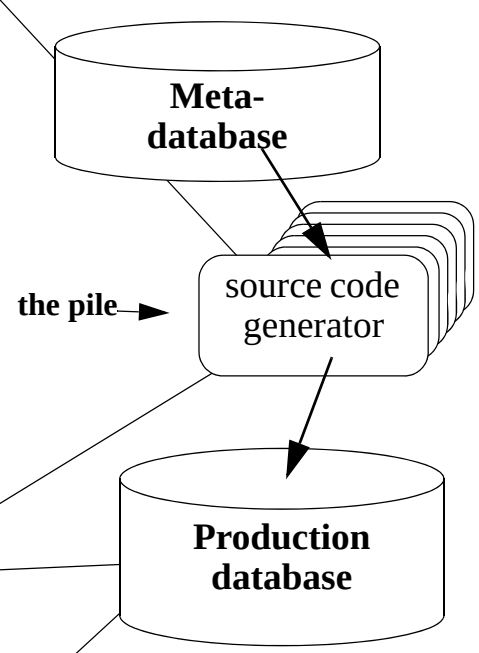


10-100 times*) Programmer Productivity 3(3)

Simplified SQL-based source code generator example in **the pile** implementing an API. These are run from automated scripts

```
select "select '/* Table declaration for class
"+c.name+" */','"
,"go"
,"select 'create table "+
         c.prefix+"_"+lower(c.name)+" ('"
,"go"
,"select '    '+a.name+' '+a.declaration+','"
,"from allattribute a"
,"where a.class_highid = "+convert(varchar,c.highid)
,"  and a.class_lowid  = "+convert(varchar,c.lowid)
,"order by name"
,"select char(8)+char(8)+')'"
,"go"
 from class c
 where c.gensqlflag = 'T'
 order by c.prefix
```

m lines of code



1 av 41 genererade källkods-”bitar” i Produktmodelldatabasen.

```
/* Table declaration for class DocumentRecord */
create table dr_documentrecord (
    approvedBy VARCHAR(32),
    authors VARCHAR(255),
    booktitle VARCHAR(255),
    cdrom VARCHAR(255),
    checkedBy VARCHAR(32),
    createdBy VARCHAR(32),
    crossref VARCHAR(255),
    definition TEXT,
    dtAdded VARCHAR(27),
    dtApproved VARCHAR(27),
    dtChecked VARCHAR(27),
```

Generation factor:

$$\frac{n}{m}$$

n lines of code

*) Over a few decades, with tens of applications and hundreds of versions, high-quality source code generators pay off much more than that.

4 Programming Language Independence, Application Markets

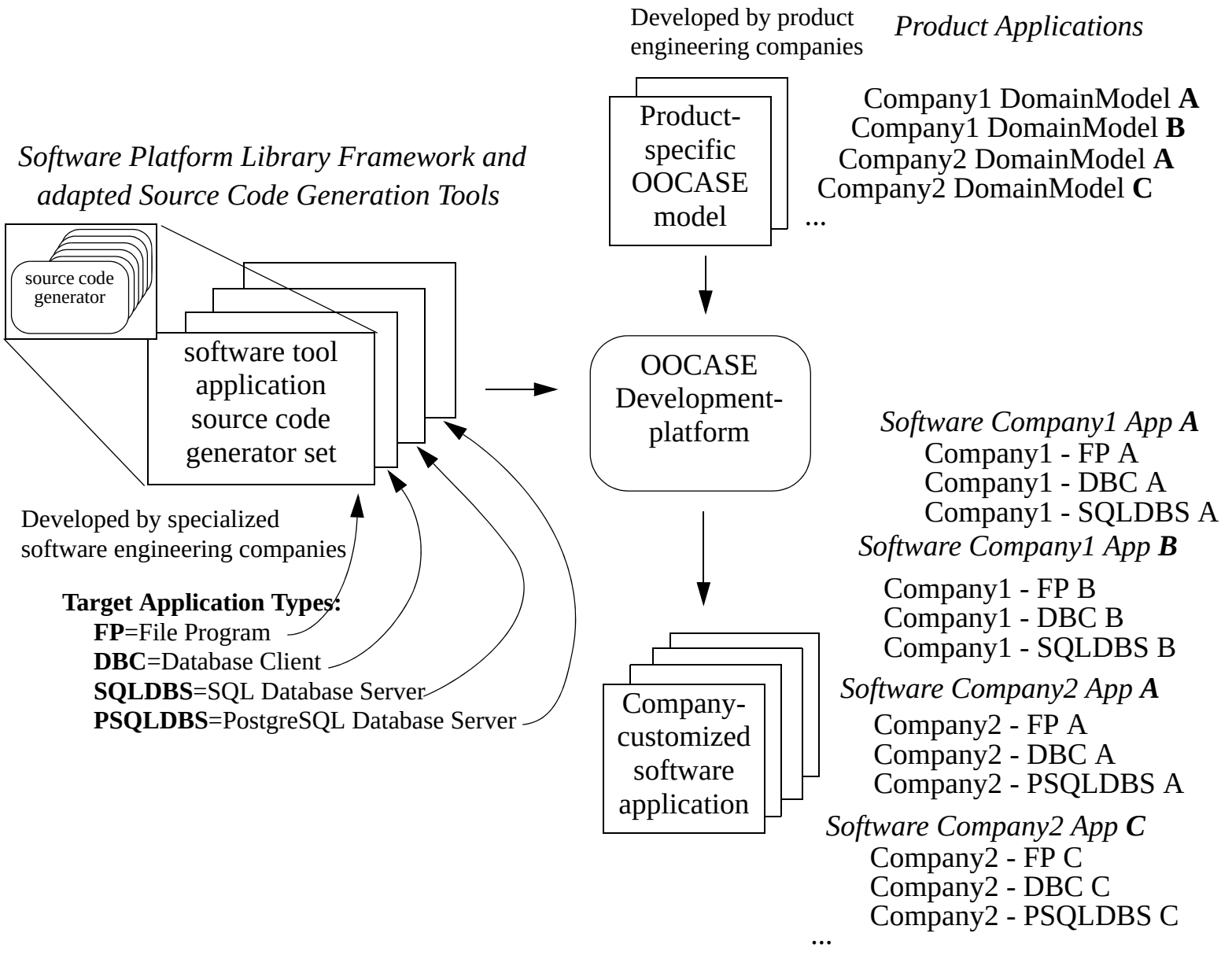
Programming Language Independence

- A DomainModel is Programming Language Independent
- Source code generators are targeted for a particular target programming language, but usable on any DomainModel
- The DomainModel can be implemented in any target programming language that has a source code generator set.

Application Markets

- The same DomainModel can be implemented in many different programs, targeted and priced for a particular market need
  - Free or open source education versions for students.
  - Simple stand-alone single-user laptop application version.
  - Client workstation application version.
  - Database server application version.
  - Mobile smartphone app version for read-only, or simpler search and update use cases.
- The object models the users develop in the applications follow a standard DomainModel. These user developed object models can be communicated between different application variants in an expanding standard enabled market eco system

Market Approach: Efficient Division and Use of Competence



5 Application Product Portfolio Reuse Advantages

Software Platform Library Framework

- Reuse standard DomainModel Modules with standard Classes, Attributes and Relationships in multiple applications and products
- Develop reusable generic manually implemented API functionality that operates on the objects within the standard Module.

Typical Standard Functionalities in cross-application shared modules

- BaseClass - superclass of all user application model objects - provides generic API that the generated DomainModel specific source code calls. Keeps the complexity down in source code generators.
- Persistence - load and store user application models on files.
- Database Client communication - generic functionality for communication of object information with a database server
- User Interface Library that handes GUI interface with the user created DomainModel specific objects in a user model.

Reuse same API in many object-oriented target programming languages

- Smalltalk, C++, etc
- Source code generators become much simpler to port to new languages

6 Global Standard for Team-Based DomainModel Module Development

The OOCASE V4 meta-model database is

- Founded on the best scientific information theory.
- Proven in industrial applications for more than 2 decades.
- Simpler, more functionally powerful for decade-spanning lifecycles of object-oriented information models than any other scaleable industry standard

No Object-Identifier Collisions in Globally Distributed Databases

- Globally unique 128 bit object identifiers for all objects.
- Each OOCASE DomainModel developing customer receives with the license a globally unique 64 bit highId, that uniquely identifies all newly created, or copies of other objects with that license.
- Each customer can create 2^63 = 9 223 372 036 854 775 808 globally unique DomainModel objects with one OOCASE License.
- Each customer can copy and reuse internationally developed and shared standard DomainModel Modules in new releases of their own DomainModels, with full traceability to the originally released shared Quality Assued standard.

7 Future Insurance 1(3)

The DomainModel of OOCASE is designed in OOCASE itself

- All customers receive with the delivery archive of OOCASE a copy of the meta-model of OOCASE itself.
- The Copyrighted meta-model of OOCASE is available for internal business use for all customers who buy a License.
- Redistribution of the meta-model of OOCASE to a customers' own customers is allowed if a contract is made with the Copyright holder.
- The OOCASE meta-model is licensed under the QAEVR License Agreement, that means that redistribution to own customers is free-of-charge, but no changes in the OOCASE meta-model are not allowed. Customer change requests for a new release of the OOCASE meta-model must be responded to in 6 months.
- Additions with other Modules of functionality that do not change the core OOCASE meta-model are allowed, and do not require any approval. Co-operation on such efforts are appreciated.

7 Future Insurance 2(3)

International Standardisation Experience for Maintenance of Standards

- Standardization of new DomainModels, whose economic or open-source software implementation infrastructure can be created with powerful tools like the OOCASE development platform and efficiently massdistributed over the internet, must rely on the expertise that create the new standards.
- Human Capital with appropriate education and contact network to users of the standard for remittance of change proposals and integration of feedback from the user community for maintaining a standard are fundamental for the Quality Assurance of any changes to a standard.
- Some private or public organization must have funding or income to maintain the educational infrastructure and contact network to enable such Human Capital to exist and have adequate working time to quality assure proposals to changes on a standard.
- The market-economy reasons for that is that uncoordinated diversification of a standard destroys a mass-market that depends on complete automation without any human attention.
- The whole market benefits, if new versions of a standard are optimized for the business needs of the whole market.

7 Future Insurance 3(3)

Future Insurance for Private Entrepreneurs and Research Institutions

- Develop DomainModel driven source code generators for your own favorite programming language platform.
- Generate an OOCASE application on that platform.
- Export/Import your OOCASE DomainModels to that platform, and develop your own source code generation capability in your favorite programming language.
- Share the DomainModels you develop with the competence and focus of your own organization in private business parner intranet, VPN networks or public repositories.
- Use your own sales channels, or the ones made available by international internet companies for making an income or promoting and enabling a shared research interest that requires massive information exchange, safe archiving and maintenance of recorded information, with your own software applications shared by peers.

Once a foundational standard is set for a market, the NEXT LAYER of VALUE ADDING APPLICATIONS can be built on that standard.

- If market actors compete each other to death by trashing standards on the "low-level", any mass-market added value coming from the possible NEXT LEVELS will never turn into solvent paying capital flows.

8 Decade Spanning Software Maintanance

Resilient foundational industry standard for information storage

- Operating System (OS): Windows, Linux
- Directory structue, Textfiles: CVS (comma/TAB separated text files)
- Relational databases: SQL92 (common denominator)
- Informations models: OOCASE (the core for classes, attributes and relationships since 1992 are the same, IEC 61360 since 1997 is still industry standard) - computer programs och APIs can be implemented in all standard programming languages
- Web browser user interface: HTML 4 (backwards compatible since 1995)

Mores Law for mass-affordable workstations

- CPU clock frequency: 1992 64 MHz, 2018 3,3 GHz, 4 cores > 200x
- Primary memory: 1992 64 MB, 2018 32 GB > 500x
- Persistent memory: 1992 128 MB, 2018 1,2 TB flash ~= 10 000x
- “More relational databases in smart-phone apps that you can imagine”

New database technology with magnitudes of higher performance

- Primary memory databases (Store the whole database in primary memory)

9 International Market for New Standard Information Exchange Formats

- Most of my students at the University and in Industry are bright people who are interested in my courses because they want leverage in their work. Besides receiving the pleasure of teaching great stuff that delivers capability to bright students, I learn a lot from their input during the courses and what they did and do later in life, where my courses may have had some positive effects.

This slide is for YOU to write

- Everything big has started from small and grown. A high-quality information exchange standard that cost-efficiently can be distributed all over the internet, can make a big difference.
- Since your knowledge domains, perspectives and life experience is beyound what I can imagine, I leave this slide to you for placing your vision here.
- For the best background on formulating that vision, I recommend you download a Student version of DocumentDictionary.V1.0.
- Study the doc/DocumentDictionary_Tutorial_V1.0.pdf
- Then test the knowledge presented by using it, by studying either the OOCASE User Manual, or the DocumentDictionary User Manual.
- The practical personal experience of having studied a document with DocumentDictionary, will give you hints how you can use it. Both for your personal education and tuning your own studying performance, and for the students or team members you work with that might need some support to get through all the background readings and manuals you need them to know, to become proficient contributors to a project development effort.
- I expecially recommend Chapter 5 Domain Model Development. in the OOCASE manual. This is key knowledge in creating the foundations for successful software projects where powerful new technology meets present business reality and the state of knowledge that your customers are at and the state of the technological infrastructure your customer uses today to support and maintain their business. If you know what they know and have, you can see what assets you can re-use in the new system.
- Efficient communication with the customer’s experts in an organization is still the fastest way to get access to all background reading neccessary to really understand what business you are buildning a new software system for. Understanding, that you can transform into added value.