

Guidelines for framework development process

V. Stanojević, M. Milić

University of Belgrade

Faculty of organizational sciences

Software engineering laboratory

<http://silab.fon.bg.ac.rs>

Outline

- Introduction to Oz framework
- Framework development process
 - Doman analysis
 - Framework design
 - Framework instatiation
 - Hot spots
- Conlusion

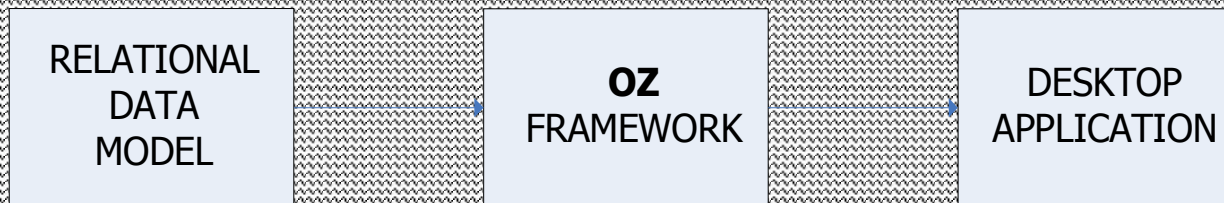
- Keywords: framework, code generation, desktop application, relational model, software patterns, c#

Introduction

- Growing need for new business software solutions
- Developing a framework which will shorten software development time
- A framework is a set of classes that embodies an abstract design for solutions to a family of problems [Johnson, 1988]
- A framework is a reusable design of all or part of a system that is represented by a set of abstract classes and the way their instances interact [Mattsson, 1999]

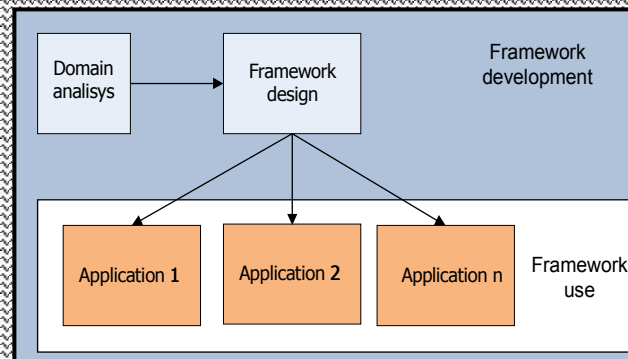
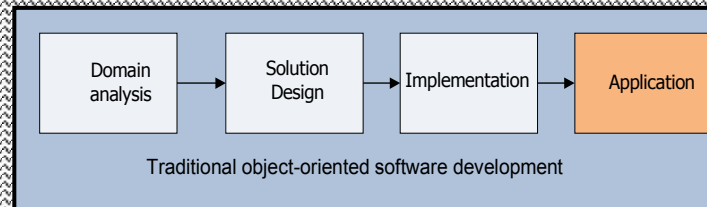
Introduction

- One of main reasons for framework development is code reuse
- Motivation for a research at Faculty of organizational sciences / Software engineering laboratory
- There is no unique methodology for framework development and documentation
- Oz framework



Introduction

- Software development vs development of framework
- Result of software development is concrete application, while result of framework development is a framework which can be used for developing several concrete applications



Framework development process

- Different approaches to framework development process
- Two different activities in framework development: development of **framework core** and development of **internal add-ins** [Bosch, 1999]
- Core exists, unchanged, in every framework instance (*frozen spots*)
- Parts of code which makes code flexible are called **hot spots**

Framework development process

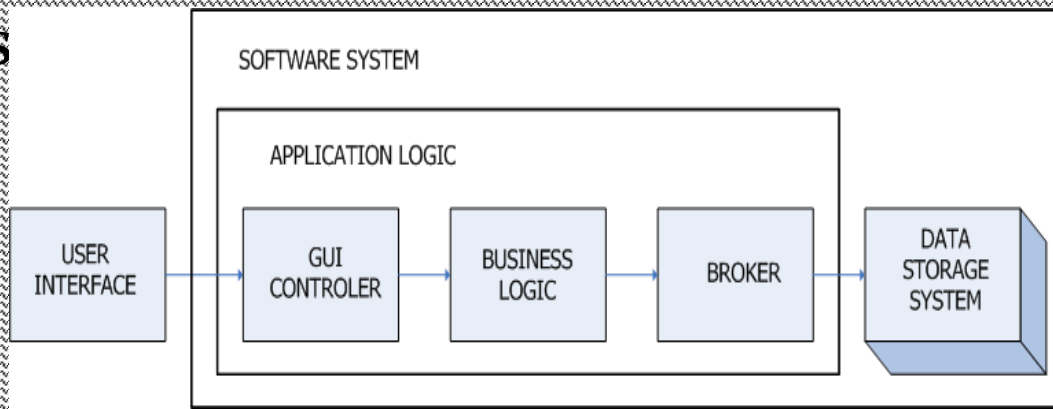
- Architecture must be reusable and very flexible
- All authors have same opinion, which we share, that use of design patterns is inevitable in framework development
- Patterns that are identified as significant in framework development [Larman, 1998; Taligent, 1994]: pattern of three examples, white box, component library, hot spots pattern, pluggable objects, grained objects, black box, visual builder, programming language tool

Framework development process

- Generating a *three tier application* which will implement CRUD operations for specific domain
- Bosch's method is evaluated as most suitable for development of the Oz framework
- Oz framework development had these phases [Bosch, 1999]: Domain analysis, Framework design, Implementation, Framework testing , Framework documenting

Oz - Framework development process

- **Domain analysis** – using pattern of three examples, which recommends that the domain analysis should be based on **three specific desktop applications**
- Objective of this phase is to define general architecture of business applications which the framework should generate (framework core and internal-add-ins)
- All parts



ized

Domain analysis

- Graphical User interface
 - General FGeneral class incorporated general behavior of every form (FRAMEWORK CORE)
 - Transformations used for form generation based on relational model must be determined (INTERNAL ADD-INS)
- GUI CONTROLLER
 - Has responsibility to do data conversion and forward call to Business logic controller
 - Abstract GUI Controller (FRAMEWORK CORE)
 - Data Transfer Object (collection of String key value pairs)

Domain analysis

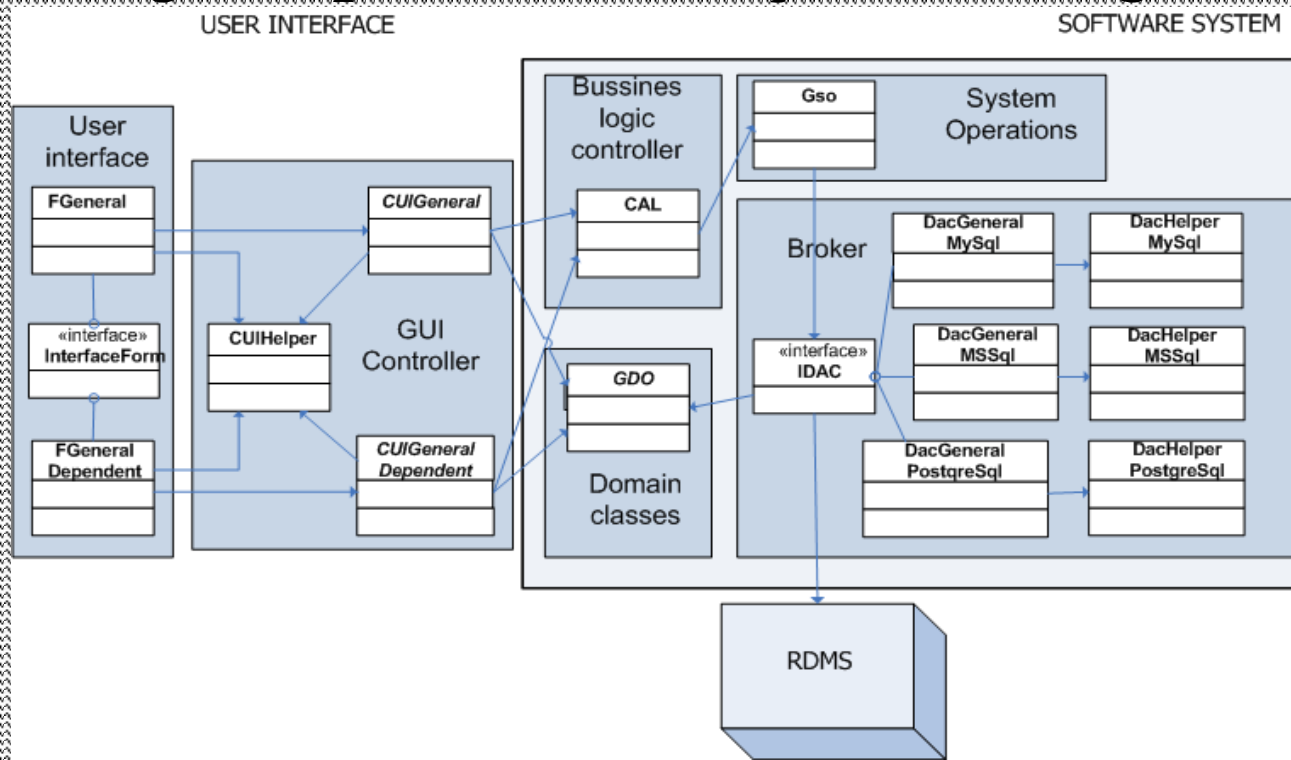
- Business Logic Controller
 - it has a responsibility to forward call to proper system operation (Framework core)
- System Operations Analysis
 - Has responsibility to call proper methods of Database broker and to do transaction control
 - Only SO for CRUD operations
 - Template method pattern (to define algorithm)
 - We have developed GENRAL CRUD SO (framework core)
 - Transformation from DTO to domain classes and vice versa
 - A good place for HOT SPOTS

Domain analysis

- Domain classes
 - General Domain Object (framework core)
 - Domain Classes are specific for every application and they can be mapped from relational database meta data (internal add-ins)
 - We have provided DTO to Domain class mapping
- Database Broker
 - Has responsibility to retrieve and change data in database
 - General database broker
 - It has general methods for CRUD operations(Bridge pattern)
 - Developer can chose one of many existing ORM tools
 - One has to make map DTO to domain classes specific for selected ORM tool

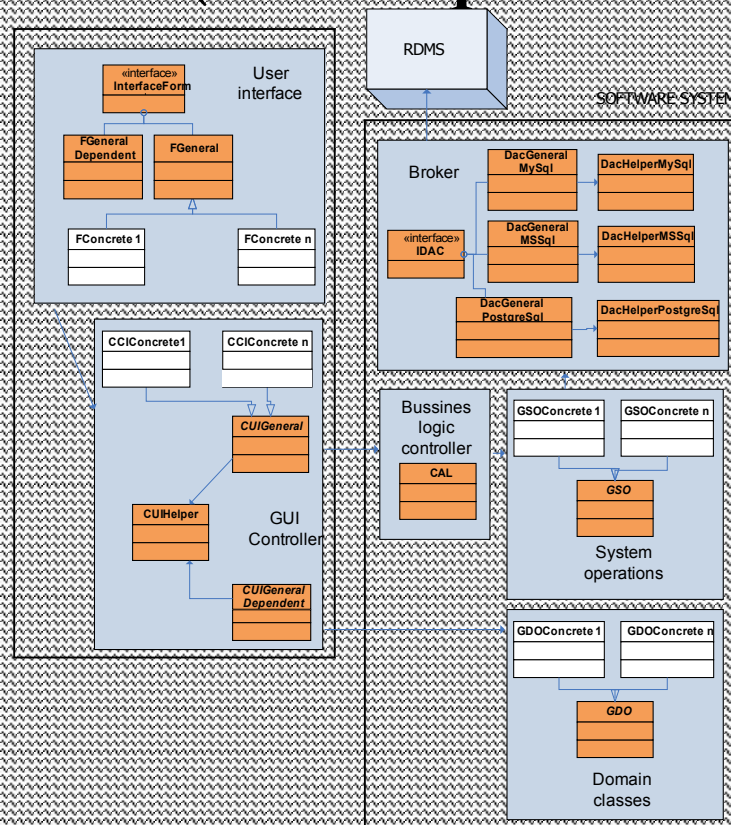
Oz -Framework development process

- *Framework design* is based on grounds of previous phase (framework core, internal add-ins)
- Building *component library* and *code generators*



Oz -Framework development process

- Relation between **core**(classes painted orange) and **internal add-ins**(classes painted white)

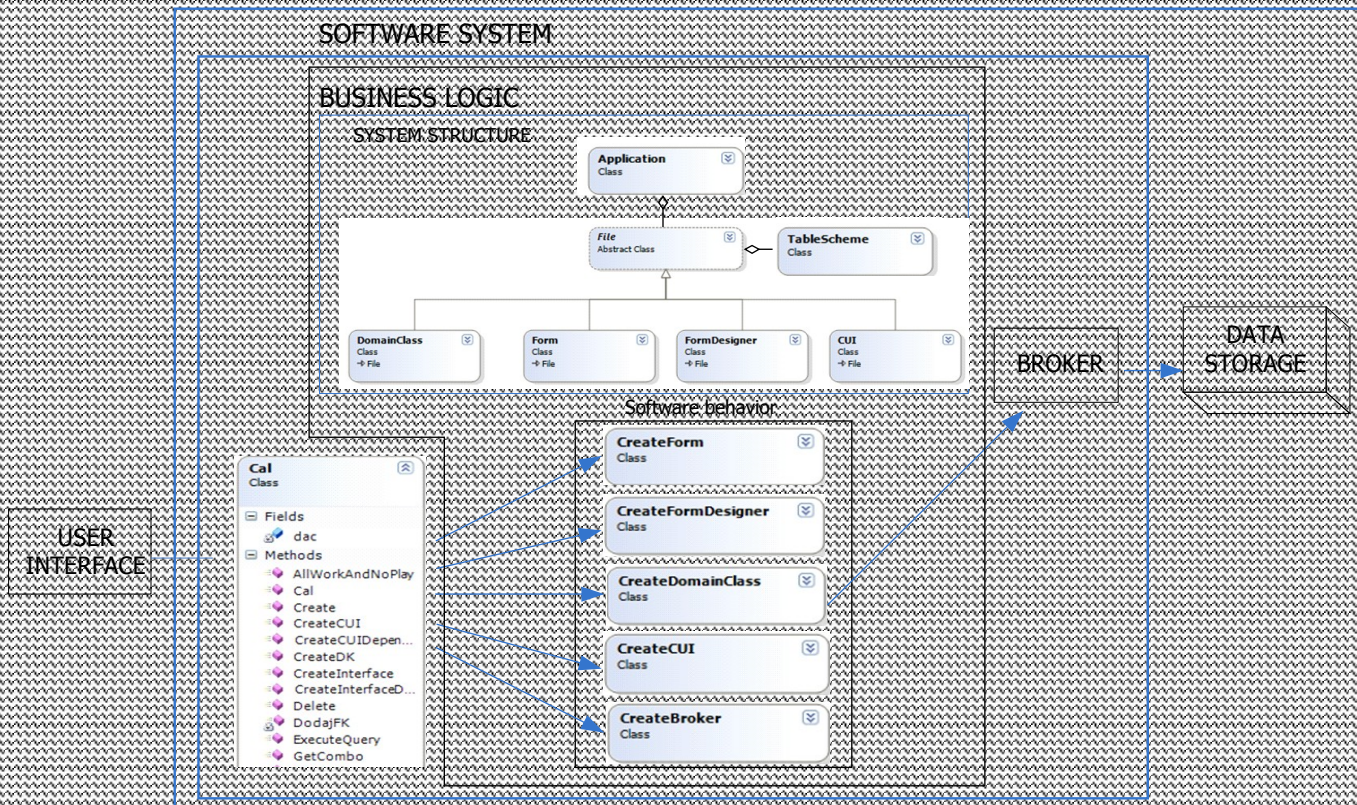


Oz -Framework development process

- Larman's software development method was used for code generator development
- Class templates were designed and developed for code generator
- Graphic form is created for every table from underlying database

Oz -Framework development process

- Oz framework code generator architecture

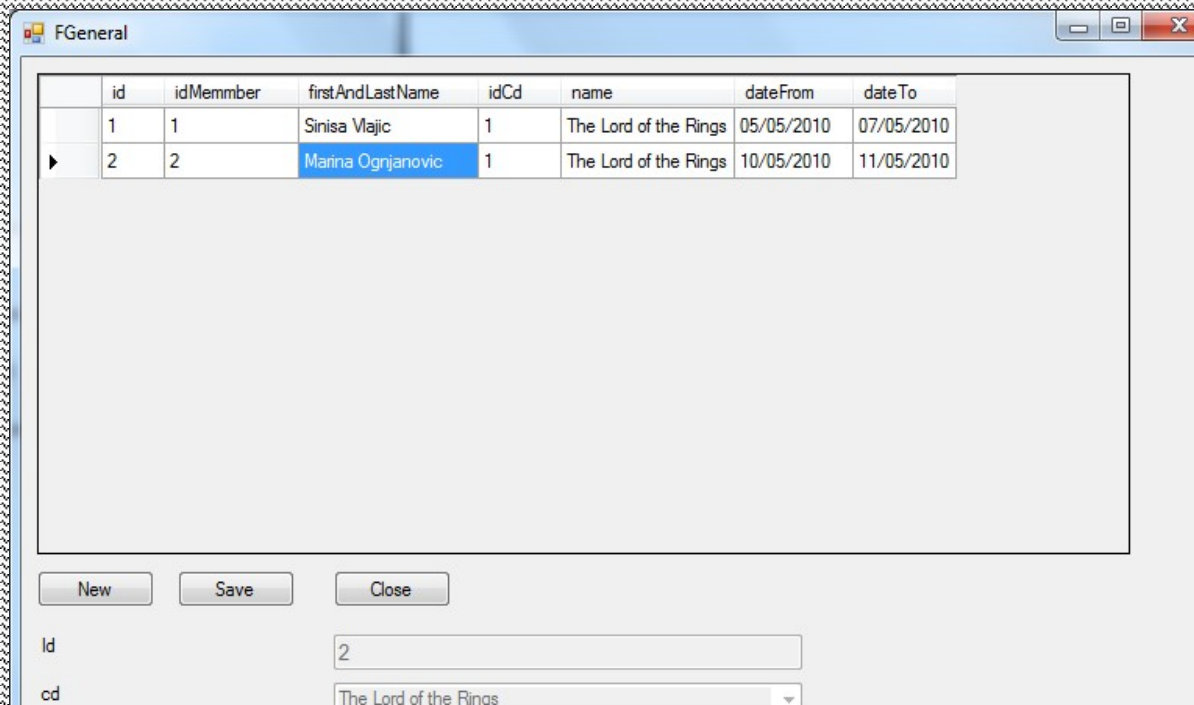


Framework development process

- Framework development process is iterative and incremental
- In most cases user will have to provide additional information during or after framework instantiation
- User should determine potential hot spots which can be made in next increment of framework development
- In order to change application functionalities one must override generic methods of system operation with concrete one, change generated ones (not recommended) or create completely new classes and integrate them in generated application

Framework development process

- Framework should enable code change only in specified (hot spots) methods of generated code.



The screenshot shows a window titled "FGeneral" with a table of data and input fields below it. The table has columns for id, idMember, firstAndLastName, idCd, name, dateFrom, and dateTo. The second row is selected, showing id 2, idMember 2, firstAndLastName Marina Ogrjanovic, idCd 1, name The Lord of the Rings, dateFrom 10/05/2010, and dateTo 11/05/2010. Below the table are buttons for "New", "Save", and "Close". At the bottom, there are input fields for "id" (containing 2) and "cd" (containing The Lord of the Rings).

id	idMember	firstAndLastName	idCd	name	dateFrom	dateTo
1	1	Sinisa Vlajic	1	The Lord of the Rings	05/05/2010	07/05/2010
2	2	Marina Ogrjanovic	1	The Lord of the Rings	10/05/2010	11/05/2010

New Save Close

id 2

cd The Lord of the Rings

Conclusion

- Generation of three tier desktop applications based on relational metadata
- Identified patterns: pattern of three examples, white box, component library, hot spots, pluggable objects...
- Bosch's method for framework development
- In domain analysis authors suggest that one should use pattern of three examples
- As a result of domain analysis components that are part of framework core and those that depends on relational metadata and as such are part of internal add-ins should be identified

Conclusion

- Framework design is done using Larman's method of software development
- Hot spots are also noted and authors gave guidelines how user can use these hot spots to change functionalities of generated applications

Guidelines for framework development process

V. Stanojević, M. Milić

University of Belgrade

Faculty of organizational sciences

Software engineering laboratory

<http://silab.fon.bg.ac.rs>