Very Rapid Applications Development on Management

Creating management software fastly

- Matthieu GIROUX - www.liberlog.fr

- Self Programmer
- Integrating Web Sites
- Management software creating
- Creating a VRAD framework

Table of contents

1.1) Gain time : Why fast ?

Now computers permits to gain time.

But creating software is more and more expensive and slow. Most of software projects are stopped.

And computers permits to automate.

So we can automate creating software.

1.2) Gain time : Ergonomics

User is always asking himself the same questions.

So Ergonomics are always the same questions.

We can think about the user before the interface is created.

How ?

With the last software, and a prepared automating work...

1.3) Gain time : Interface

Creating each interface entirely :

- Creates ergonomics problems
- Looses time : To create and correct
- Looses in quality : A lot for nothing

It is known : A lot of softs are slow and badly made, because there some copy-paste.

1.2) Bad example
Some programming API

A API is a tool package to program

Management API needs :

- To create source code hardly manipulated
- To program to remake that exists already

A Engineer can transform some Management API de gestion to create the interface of management software from some existing files of VRAD API.

1.3) Good creation of a library

Story of framework creating

- At the begining we create some functions units
- So we use and inherits components
- Then we create components packages
- We open our library to other libraries
- So we automate some package in a library
- The library needs no code or less

1.4) History : Rapid Application Development

A software is made by :

- Job part : That is asked by the customer
- Computing part : The software

With with RAD or other methods we :

- Mixed the job part with the computing part
- Remade the software entirely

The job part must be kept.

1.4.1) Gain time : Faster

If we centralise each programming of software :

- We anticipate the useless of software
- The software is quality based : Tests are used
- We participate with the projects we use
- We create its own framework

If we centralise we create a very fast development framework.

1.4.1) RAD Components : Faster

RAD Lazarus Components can be installed with a simple mouse clic.

The Objects' Inspector will fill RAD Component quickly.

Components can make a VRAD Engine easily, based on Business Process Management model.

1.5) History : Rapid Application Development

We see that the customer must know how work a software but he does not know to manipulate computer.

It is possible to pass fastly the computing part creating the software from the user will.

The customer must always know :

- How to make a software (RAD, VRAD, etc.)
- If he has the hand on the created software

1.6) History : Rapid Application Development

Rapid Development tools have been created to getting more and more programers :

- VISUAL BASIC, DELPHI, LAZARUS

Now some frameworks can be included to them. They can keep job part :

- GLADE GTK for interfaces not linked to data
- JAVA LEONARDI and WAVEMAKER for managament
- GAME MAKER for games.

2.1) Management software

A management software is :

- An enterprise software
- Administration of web site
- Accounting of enterprise

We see that we can model easily a management software.

2.2) Management softwares

A Management software is :

- A link to data system
- Some relationships beetween forms
- Some statistics, some arithmetic
- Some maps, or other plugins

All of that is defined and automatable.

A VRAD software automates this systems.

3.1) The Multi-plateforms

The maker can use is own framework.

The multi-plateforms is :

- Be self-sufficient of computing system
- Be self-sufficient of used framework ?

It is always possible to change of framework with data.

It is possible to be more self-sufficient with VRAD.

3.2) Rapide Application Dévelopment (RAD)

- Creating visually some software
- To gain time with programing
- To create easily a sofware

Most of RAD tools don't automate or less an enterprise management.

The Very Rapid Application Development permits to getting RAD faster for management servers or other modelisable interfaces.

2.5) Creating an interface with passive files : VRAD

It is now possible to create a management interface with Model Driven Engeenering.

A passive file containing user part is read and creates the interface from VRAD engine.

- GLADE GTK permits to create an interface, only actions are coded.
- LEONARDI permits to create a management interface from passive files.
- LIBERLOG owns a RAD framework with a VRAD Engine.

2.6) Creating an interface with passive files : VRAD

Passive files :

- permit to create objects' models.
- Are the customer demand.
- will automate the showing.

It is possible to create themes of interfaces. So user can choose the showing.

4.1) Good points on Very Rapid Application Developing

The Very Rapid Development permits :

- To prevent errors from occuring
- To create only the models at the end
- To gain some time with the creation
- To be self-sufficient
- That the programer thinks functionalities

The source code will reusable, centralized, user friendly, integrated more easily.

4.2) Rapid Development vs Command Line

Example : Creating a simple form

A centralised code used with some copy-paste

- 3 days and it is not always finished

Same making with a RAD tool

- Analysing ½ day and ½ day of creating
- The component automatee some creating
- The form is usable with less tests

4.3) Passive files of VRAD vs RAD

Passive files :

- Permit to modelise the Job Part in its mind
- Can be created from analytic or data
- Can be slef-sufficient from all used framework
- Are defined a must be evolutive
- Permit to creating some more interfaces
- Permit to think functionality

The analysis of ½ day qui creates the software.

The analysis is always same as created software.

4.4) Passive files of VRAD vs RAD

With passive files we :

- Think functionalities and Job Part with models
- Determine what can be done quickly
- Determine what is not modeled
- Create plugins for what is not made
- Know where we are going

Once the software is created you can create other types of interfaces with VRAD framework.

4.4) VRAD Quality

With a VRAD engine :

- We gain in time and be more agile
- We facilitate the making of future softwares
- Test only the engine, not the created interface
- The analysis is the software
- The maintenance is centralised
- The programer goes to the fundamental

5.1) Creating some VRAD plugins

Creating some VRAD plugin :

- Will be integrated with passive files
- Will be faster if you use a RAD IDE
- Will be acquired once the plugin created
- Require to create the plugin on other EDI
- Will be modeled in the analysis

6.1) VRAD LEONARDI GPL LGPL

This VRAD library permit to :

- Create passive files with analysis
- Create Software with passive files
- Creating the computing part with plugins
- Doing Reverse Engeenering from data
- Create software for both web and not web

6.2) VRAD LEONARDI GPL LGPL

With LEONARDI can be in the GUI:

- Managing with forms
- Sorting, filtering, searching, composing
- Printing, exporting, importing
- Creating statistics, trees, tables
- Creating charts, maps (not free)
- Creating plugins linked to passive files
- Learning easily with french community

6.3) VRAD LEONARDI GPL LGPL

LEONARDI can make your:

- Supply Chain Management (SCM or SCM)
- Customer Relationship Management (CRM)
- Monitoring, Network Administration ...
- Requirements: network equipment ...
- Information Communication System (CIS)
- Command Support System
- Integrated Management Software (ERP)
- Managing Repositories
- Geographic Information System (GIS)
- Inventory Management

It is possible to create a prototyping fastly.

6.4) LEONARDI – RESTRICTION

LEONARDI library is free:

- To create commercial software
- Using Free Data Systems

- The data links are paid when paying Data Systems.
- Mapping and GANTT are payed

6.5) LEONARDI – Facilities

Making LEONARDI better :

- Creating some plugins and free components
- Some similarities with JELIX JFORMS files
- Using management API with LEONARDI files

6.6) LEONARDI et JELIX JFORMS Similar frameworks

LEONARDI analysis permits to :

- Create some software
- Using passive files of forms
- That creates the software

It is in theory possible to translate some files between LEONARDI and jelix JFORMS.

We are so independent of any tool.

7.1) JELIX – LGPL

This library permits to :

- Create a management software with JFORMS plugin
- Create computing part with JELIX plugins
- Create a software or web site

Advantages

- A lot of plugins for WEB portal
- It is Possible to converte the XML files

It is possible to transfer all LEONARDI analysis.

7.1) WAVE MAKER – APACHE LICENSE

This library permits to :

- Create easily a management software
- Create a software or web site

Advantages

- User friendly

7.1) GAME MAKER – COMMERCIAL

This library permits to :

- Create a game easily and quickly

Advantages

- Programming is not needed
- Not necessary to adapt on every platforms

Misses

- We depend on the software
- We must pay the evolution

8.1) Why using EDI RAD ?

- Rapid changes.
- The components are quickly put in place.
- The component structure is homogeneous.
- Easy maintenance.
- Centralization and individualization sources.
- Do not create unnecessary.
- Sorting according to the computing part.

8.2) LAZARUS

Advantages

- Opened and friendly project.
- On WINDOWS LINUX UNIX MAC-OS BSD.
- A lot of DELPHI free components.
- Fast Execute because not traduced (no JAVA).
- An executable by platform.
- Creating fastly if good programing.
- Creates web or not web software.

8.3) LAZARUS

Bad points

- Heavy executables.
- Began on 1999 : (Not fully DELPHI compatible).
- Need to rewrite WINDOWS APIs.
- Graphic part remade so 98 % compatible.
- Traduced components have less properties.
- You use multi-platform units.
- More powerfull on WINDOWS, and so LINUX.

8.5) How to create RAD component?

How to work?

- Easy component.
- Scalability.
- Portability.
- Interoperability with other components.
- Anticipating on the structure of the component.
- Methods and variables with appropriate English.

8.6) The power of LAZARUS

LAZARUS is a RAD EDI which complains :

- Framework of LIBERLOG.FR.
- Data System.
- Visual executables on WINDOWS,LINUX,MAC.
- Embedded on some mobile phones.
- Creating WEB with components.

9.1) LIBERLOG FRAMEWORK

XML FRAMES

- Creating some management software.
- With RAD management components.
- Creating fastly some simple forms.
- Possible use of some framework like LEONARDI.
- It will be possible to create some embedded softwares.

9.2) Why a framework ?

The used framework :

- permits to creating some interfaces.
- Permits to be self-sufficient of the software maker if it is free with all software sources.
- Can centralise Job Part if demanded.

If Job Part is not centralised so we see mistakes between analysis and software creating.

9.3) Creating a VRAD framework

- MICROSOFT is an organization that does not want the self-sufficiens of its customers for its future tool VRAD.
- Only end customers, little or middle enterprises see the interest of VRAD Management.
- We must use the open source available and create a single file VRAD format.
- The creation of a free expertise VRAD permits to retrieve a part of the software business.