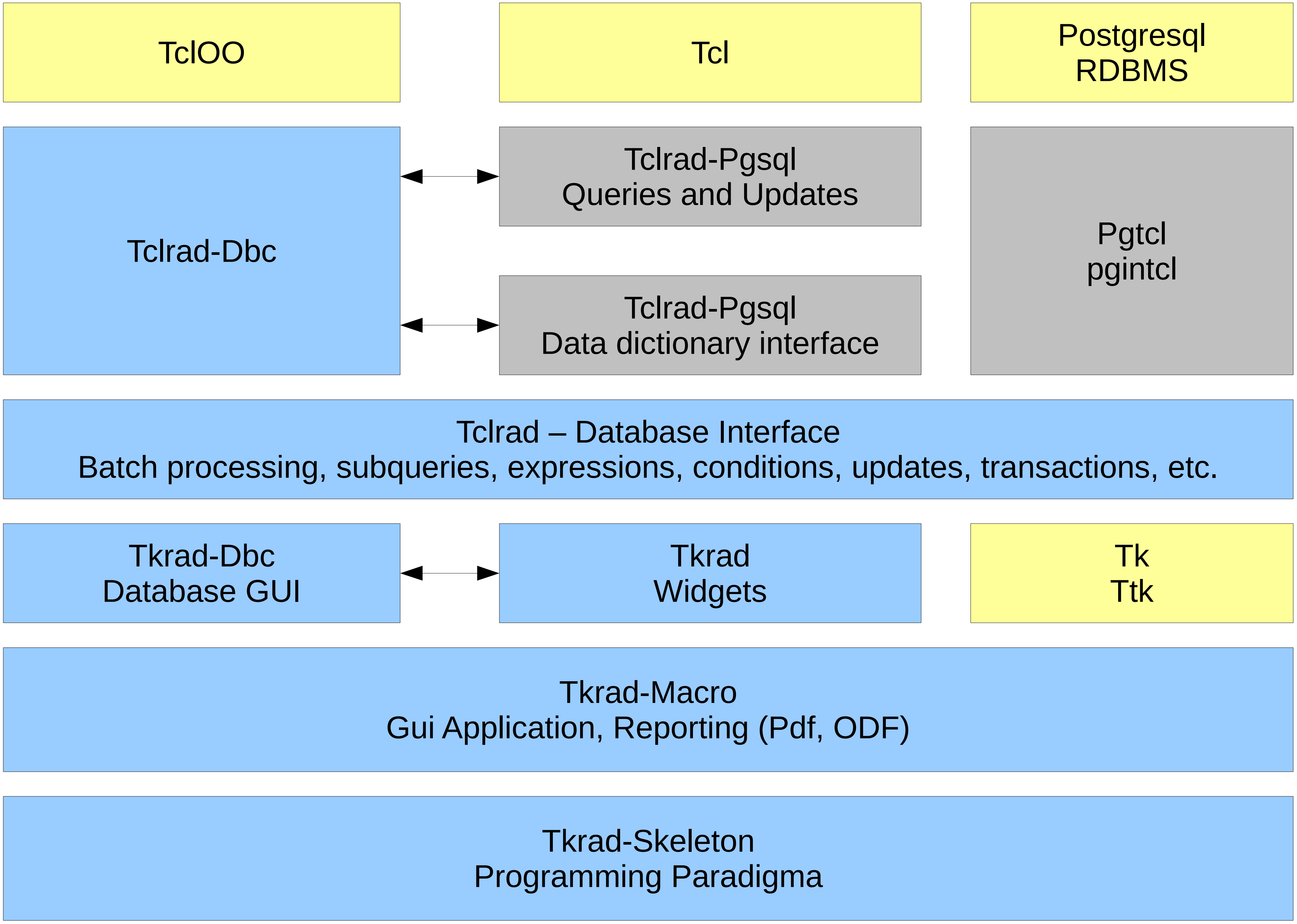
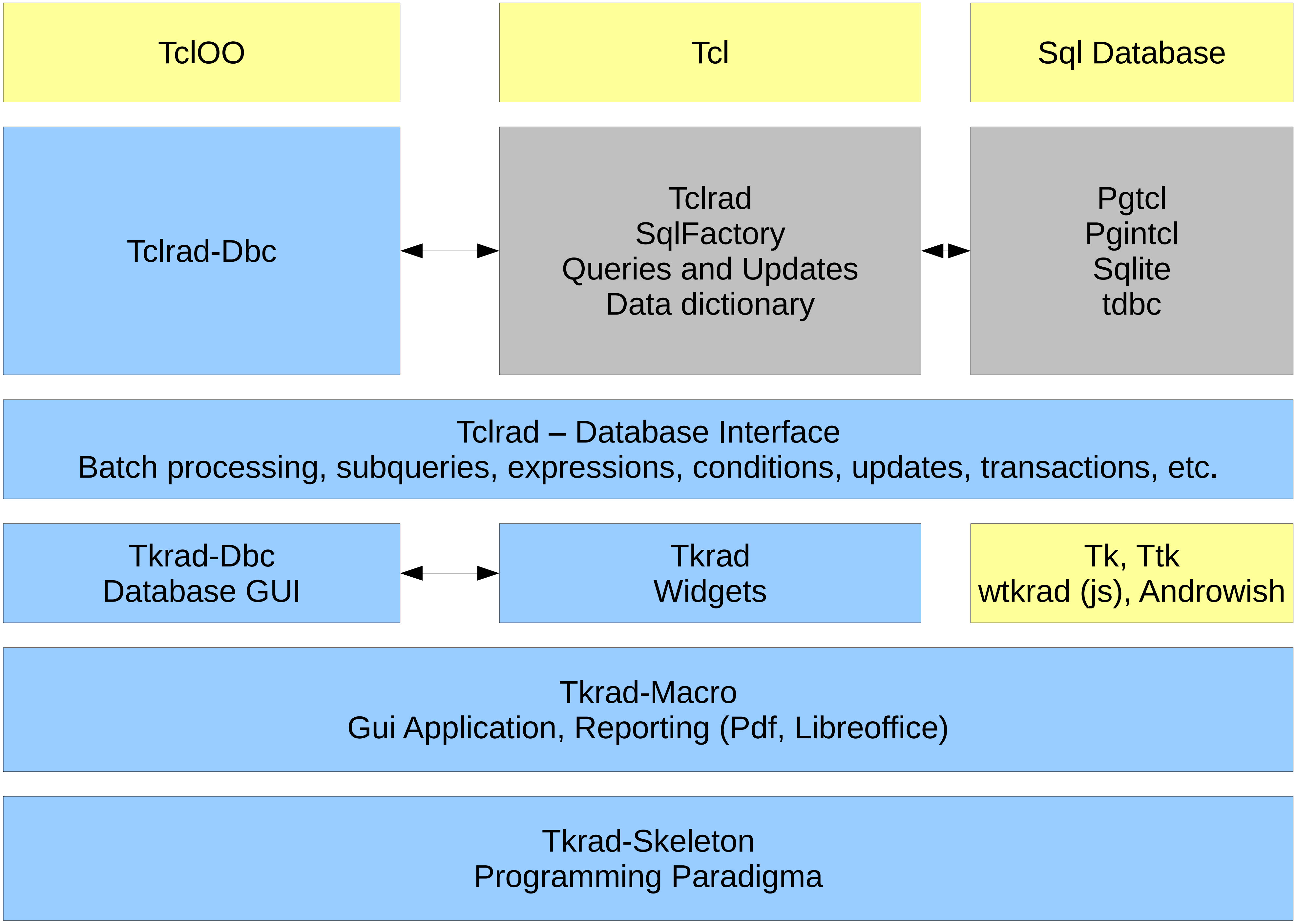
Tclrad

Rapid Application Development System
a proposal to the Tcl community

Born in Metodo, Modena

- Company
  - 20 employers (programmers, helpdesk, ...)
  - IT company in a holding of 650 employers
  - 1200 linux desktops (including customers)
  - 360 linux servers (including customers)
  - 6900 tcl modules, 1.410.000 lines of tcl code
  - Custom Linux distribution (itux, a fedora/centos spin)
  - Developing legacy applications (tcl and cobol)
    - Database isam (FAIRCOMS) powered by sql
    - Database sql (postgresql and sqlite)


TclOO
Tcl
Postgresql RDBMS
Tclrad-Dbc
Tclrad-Pgsql Queries and Updates
Tclrad-Pgsql Data dictionary interface
Pgtcl pgintcl
Tclrad – Database Interface
Batch processing, subqueries, expressions, conditions, updates, transactions, etc.
Tkrad-Dbc Database GUI
Tkrad Widgets
Tk Ttk
Tkrad-Macro
Gui Application, Reporting (Pdf, ODF)
Tkrad-Skeleton
Programming Paradigma



TclOO
Tcl
Sql Database
Tclrad-Dbc
Tclrad
SqlFactory
Queries and Updates
Data dictionary
Pgtcl
Pgintcl
Sqlite
tdbc
Tclrad – Database Interface
Batch processing, subqueries, expressions, conditions, updates, transactions, etc.
Tkrad-Dbc
Database GUI
Tkrad
Widgets
Tk, Ttk
wtkrad (js), Androwish
Tkrad-Macro
Gui Application, Reporting (Pdf, Libreoffice)
Tkrad-Skeleton
Programming Paradigma

Technologies

- Database
  - Postgres
    - Pgtcl (c extension) or Pgintcl (pure tcl)
  - Sqlite3
    - Data and application support
    - Also present in Postgres deployment
      - data transfers
      - tables content and resources delivery
  - Tdbc
    - Postgres and Sqlite3
    - Odbc (for Cobol/Faircom Ctree isam integration)
  - Huge use of large objects or blobs, depending on database capabilities

Technologies

- TclOO
  - Object management system
- Icons
  - Tango icons included

Technologies

- Tdom
  - Web services (i.e. ECB exchanges)
  - Soap (i.e. EC vies vat number checks)
  - LibreOffice interface
- Tls
  - Internet and Intranet environments
- Nodejs
  - Tkrad browser interface
- Androwish
  - almost nothing has been done, it just works

Technologies

- Pdf4Tcl
  - Reporting system
- LibreOffice
  - When user wants to maintain own forms
  - As a pdf generator
  - Mandatory when pdf/a is required
- Tcllib
  - Mail and Ldap
  - Internet standard file format (mime, base64, etc)
  - Pki infrastructure (sha1, sha256, uuid)

Tclrad

- Development
  - Applications never speaks SQL, just tclrad
  - Applications never speaks Tk, just tkrad
  - Use also existing databases, data dictionary is parsed on the fly when application initializes the specified object
  - Command line development
  - Command line deployment

Tclrad

- Delivery
  - Linux 32/64, MacOs/64, Android and Windows 32
  - wtkrad is tested on Firefox and Chrome
  - some days of work for languages support (msgcat)
- Maintenance
  - Release cycle written in Tcl
  - fossil in the future ?
- Support
  - Context Ticketing System

Demo Application: Goals

- An order maintenance application
- Multicompany
- Multidatabase
  - Postgres
  - Sqlite
- Customizable
  - A single customer should be able to see the customized version of a program

Demo Application: Structure

- Base system
  - Geo informations
    - State, County, Town
  - Companies end company's customers
- Order System
  - Products and orders
- Customize
  - Standard order maintenance is customized for a single deployment

Demo Application: Delivery plan

- projects
  - libraries, programs, resources (i.e. openoffice templates), database catalog
- dependency
  - tables priorities (no county without state)
  - orders needs customers (the order application needs the customer data entry module)
- application is delivered as a mix of one or more subproject, mixed in a mega project named release
- Release project defines also how subprojects fire themselves into distribution

Demo Application: Projects

- tclrad, the framework
  - copied as is into the application development tree
- runtime, the base applications
  - depends on tclrad
  - delivered as end-user application
- orders
  - depends on runtime and tclrad
  - delivered as end-user application
- mycustomer
  - depends on orders, runtime and tclrad
  - delivered as customer level application

Demo Application: Release

- release project
  - the tree that is delivered to customers
  - each installation has all or a subset of the application's projects, depending on the customer
  - delivery and updates are generated from this tree
- A look to packages/profile
  - It profiles the distribution
  - The environment TCLRAD_CONNECT
- A look to etc/sqlite.con and etc/postgres.con
  - The connections to the database system
    - Sqlite and postgres

Demo Project Runtime

- Fire the shell into a project
  - <RAD>/bin/radproject runtime
  - <RAD> is the root development, here /tcl
  - Projects are assumed to be in <RAD>/prj
  - Environment
    - PRJ defines project location
    - TCLLIBPATH defines the library search path
    - TCLCODEPATH defines the modules search path
    - TCLRAD_PROJECT is the projects root directory
- runtime is the master project
  - the master project contains the directory catalog
  - it defines the application's database schema

Catalog's domains

- directory catalog/domains defines columns common to more than one table
  - customer code is a domain and its structure is defined once

Catalog's dictionary

- directory catalog/dictionary defines
  - context
    - STATIC (common to all installed systems)
      - state, county and town are the same everywhere
    - ENV (common to all database instances on a system)
      - large objects addressing, special table lo_root and xml_root
      - templates, special table lo_templates
    - SLOT (the application data)
      - customer table
      - special table, lo_report
  - dictionary project ownership
      - runtime does not need product
  - priorities
      - customers table must be created after company table

Catalog's tables

- directory catalog/tables defines table's related resources
  - tables/state/050functions.pre
    - defines functions to be created BEFORE the table
  - tables/state/table.def
    - defines the table state
- tables county and town
- tables company and customer
- table product and order_header
- tables order_detail
  - tables/order_detail/050view_orders.pst

Browsing Catalog

- catalog/dictionary/tclrad
  - the context, project and priority file
- catalog/domains/tclrad
  - the domain file, defines the fields shared by more than one table
- catalog/tables/*
  - define the tables
- Browse it

Project's binary tree, bin_prog

- module.tcl, is a main tcl program
- tcllib, contains libraries (*/pkgIndex.tcl)
- tclpkg
  - foreach project, the file project_name.lib is the mega pkgIndex of the projects
- projects_image
  - foreach project, file project.lib contains a computational description of the project's objects
  - the update process trusts these files to decide what has to be **upgraded or retired**

Project's binary tree, bin_shell

- bin_shell
  - contains batch commands and utilities
  - tipically, these commands are not used by end-user

Compiling catalog

- browse binary tree
- dbcompile
    - Assembles the library dbCatalog
        - dbCatalog resides on source library directory tcllib
            - dbCatalog.pkg, the library profile
            - dbCatalog.dic, a zip file containing the catalog tree
    - Compiles the library
        - library dbCatalog, like other libraries, is compiled and commited to the bin_prog/tcllib/dbCatalog directory, under the project binary tree
- dbCatalog
    - look at a standard library definition

Releasing catalog

- put_project
  - This command commit the binary project's tree to the release project
  - executed when all modules and libraries are committed to binary tree
- ptcl module_name
  - compiles module from tclprog directory
- ptcl library_name
  - compiles library from tcllib directory
- prjcompile
  - compile the whole project

Testing Catalog

- by default, the demo application works with sqlite
  - check the environment TCLRAD_CONNECT
- pgprofile change the connection to postgres and run the command
  - dbtable -table state (test sqlite)
  - pgprofile dbtable -table state (test postgres)
- dbinfo connect and parse db dictionary
  - dbinfo -table state
- dbsql is a wrapper to the database's appropriate command line tool

Testing catalog

- dbtable -table order_header
  - the concept of the function tclrad_sequence
  - the function delivery_year
    - browse sqlite database
    - select from postgres database
- dbtable -table order_detail
  - selecting the view in sqlite
    - echo “select * from order_view limit 2;” | dbsql
  - selecting the view in postgres
    - echo “select * from order_view limit 2;” | pgprofile dbsql

Generating the database

- tclrun tkradcatalog
  - tclrun is the tclsh runtime wrapper
  - it searches the TCLCODEPATH to find the module
  - tkradcatalog is the database maintenance module
- pgprofile tclrun tkradcatalog
  - the same on postgres
  - here context is much more visible
- catalog by project and context
  - tclrun tkradcatalog -project runtime
  - tclrun tkradcatalog -project runtime -schema SLOT

Library baseLib

- browse module
  - baseLib.pkg
  - baseLib.tcl
    - setup an application
    - application opens and run the connection object
  - baseState.tcl (mantainer and lookup)
  - baseTown.tcl (mantainer, lookup and foreign)
  - baseCompany.tcl
- compile library baseLib with ptcl
- compile whole project with prjcompile
- commit the project with put_project

Modules

- sqlExamples.tcl
  - some examples on sql factory structure
- lockExamples.tcl
  - how columns are shared between objects
- baseMnt.tcl
  - driving tclrad objects using alias
- compile module with ptcl
- run module with tclrun

Compile the menu

- compile the application's menu
  - look tclprog/tkmenu-main.tcl
  - look tclprog/runMenu.tcl
- compile the program: ptcl
  - ptcl runMenu
    - it is scripting, the action is symbolic. We tell to the system that program is ready to be commited on release
  - ptcl tkmenu-main.tcl
    - committed as tkmenu/main.tcl

Demo Application: Release

- Compile and commit all the projects
  - fire into release project
    - <RAD>/bin/radproject release
  - distcompile
  - run the application
    - tclrun runMenu
  - run the application customized
    - tclrun runMenu -custom mycustomer
    - see the order maintenance program
    - Printing orders
    - Reporting

Release deployment tree

- bin_shell
- bin_prog
- install
  - distrib
    - contains the distribution tree
  - file YYYYNNNN are the updates lists
- generate the distribution
  - cd $PRJ/install/distrib
  - sh make.sh

Upgrading the application

- Upgrade process
  - The files to be delivered are computed using the timestamp of the release file
  - The file list is then splitted into projects using the normal distribution algorithm
  - The updates of the catalog are submitted as a project named 'updates', customized under bin_prog/updates
  - The projects that need to be upgraded, on a customer point of view, is defined by the content of the projects_image directory
- The files are then distributed as a subset of the full application

Upgrading the application

- Fire into updates project
- Browse $PRJ/sqlfix
  - sqlfix -release 20140001
  - put_project
- Fire into release project
  - change path to $PRJ/bin_prog/updates/20140001
  - tclrun tkradupdates -sqlplay database.zip

The catalog macro language

- Used on
  - delivery to build or recheck (reinstall) the appropriate catalog
  - updates to make appopriate database change
  - distributing STATIC context tables
- Catalog
  - each update can contain the full catalog, to maintain the exact time context of the fixes

The install/upgrade logic

- Context
  - STATIC and ENV are upgraded one time
  - a loop on each SLOT is then executed
  - each schema containing a table named 'release' is assumed to be a tclrad schema, to be maintained
  - Contexts are assumed from the declaration inside the connection profile (TCLRAD_CONNECT)
- All the contexts can be deployed into a single schema (i.e. sqlite 'main')

Deploy to web

- Fire into release project
- execute command nodestart
- check /tcl/node/nodesite/site
- browse http://127.0.0.1:6666

Deploy to mobile

- Used only in mobile selling context
- Initially worked on China imported tablets with Linux Fedora installed
- Now we are testing Androwish
- Dedicated hardware

A look to a real world deployment

- Browse
  - bin_prog
    - tcllib, updates, packages_image, tclpkg
  - catalog
- Use the application
- Install the application
  - download installer
  - setup applications
  - run application

A look to a “real world” deployment

- Use a real world database
- Tclrad shows as
    - Tcl can also be an alternative to Cobol and Rpg

Developers

- Piera “Vampiera” Poggio
  - piera@metodo.net
  - wtkrad designer and developer
- Franco Violi
  - franco.violi@metodo.net

Question time