

Ratcl & Vlerq

Jean-Claude Wippler

EQUI 4
software

The Netherlands

Tcl «sucks» at data

- * yes: lists & arrays, but structures? booh!
- * so? use a database, dummy!
 - * different language, different mindset
 - * 2 worlds: extract, use, apply changes
- * foreach? mix with Tcl code? traces?
- * why is writing an app so much work?

Spreadsheets

- * ya' build things and they just come alive
- * you ACT, the system REACTS - no coding!
- * but a grid of cells is an awful mess ...
- * I want more structure, relational please
- * ... and LIVE DATA, so I can spend time on logic, not on "action A affects X, Y, & Z"

Go away, UI!

- * visual design is necessary - Tk & Web
- * a lot of code is manual coding of effects
 - * why spell out so many details?
- * I want DATA-AWARE widgets
 - * which track changes to data & styles
- * Model/View/Controller: dynamic views!

Ratcl

- * Relational Algebra (and more) for Tcl
- * table-like data model, called “views”
 - * sub-views can nest to any depth
- * select, project, join, product, group, sort
- * union, intersect, except - concat, slice, ...
- * syntax designed for Tcl, with “pipelines”

Ratcl - benefits

- * the "view" mindset

- * set-wise convenience

X: \$X extend C:I { 2+E*3 }

	D	E	F
0	a	100	aa
1	b	20	bb
2	c	3	cc

	D	E	F	C
0	a	100	aa	302
1	b	20	bb	62
2	c	3	cc	11

T: \$T group {A B} G

	A	B	C	D
0	a	b	c	d
1	a	b	e	f
2	b	c	e	f
3	e	d	c	d
4	e	d	e	f
5	a	b	d	e

	A	B	G
0	a	b	CD
			0 c d
			1 e f
			2 d e
1	b	c	CD
			0 e f
2	e	d	CD
			0 c d
			1 e f

```
$T rowid N | where {B > "b" || C == "c"} | project {D B N} | sort
```

- * vectors out-run Tcl, even its C prims

- * it's a DB: instant open, fail-safe commit

Rasql

- * SQL is a standard, efficient, simple
- * Ha ha ha!
- * everyone knows SQL, it's a safe choice
- * so true ... so was COBOL
- * Rasql - thin layer on top of Ratcl
 - * full SELECT, probably nothing else

Say "vlerq"

- * an agonizing decision:
 - * rhymes with "flair"
 - * not "burp" or "jerk"
- * repeat after me:
 - * flair-ck!
- * vlerq, vlerQ, whatever

Take **v**ectors

Add a **l**anguage

Make it **e**mbeddable

Use the **r**elational model

Include a **Q**uery mechanism

vlerQ

a concept ~~car~~ software model

It gets worse

- * you think Vlerq was a weird name?
- * how about ...
 - * Thrive - embedded Vector Engine VM
 - * Thrill - Low-level Language for Thrive
 - * Ratcl - Relational Algebra for Tcl
 - * Rasql - Relational Algebra based SQL

Vlerq - research project

- * Ratel will be its first “tangible” result
- * The ultimate goal is DATAFLOW
 - * make data-aware GUI widgets feasible
 - * change propagation & traces
 - * networked “backplane” - Tequila
- * Tcl is a great fit - for several reasons

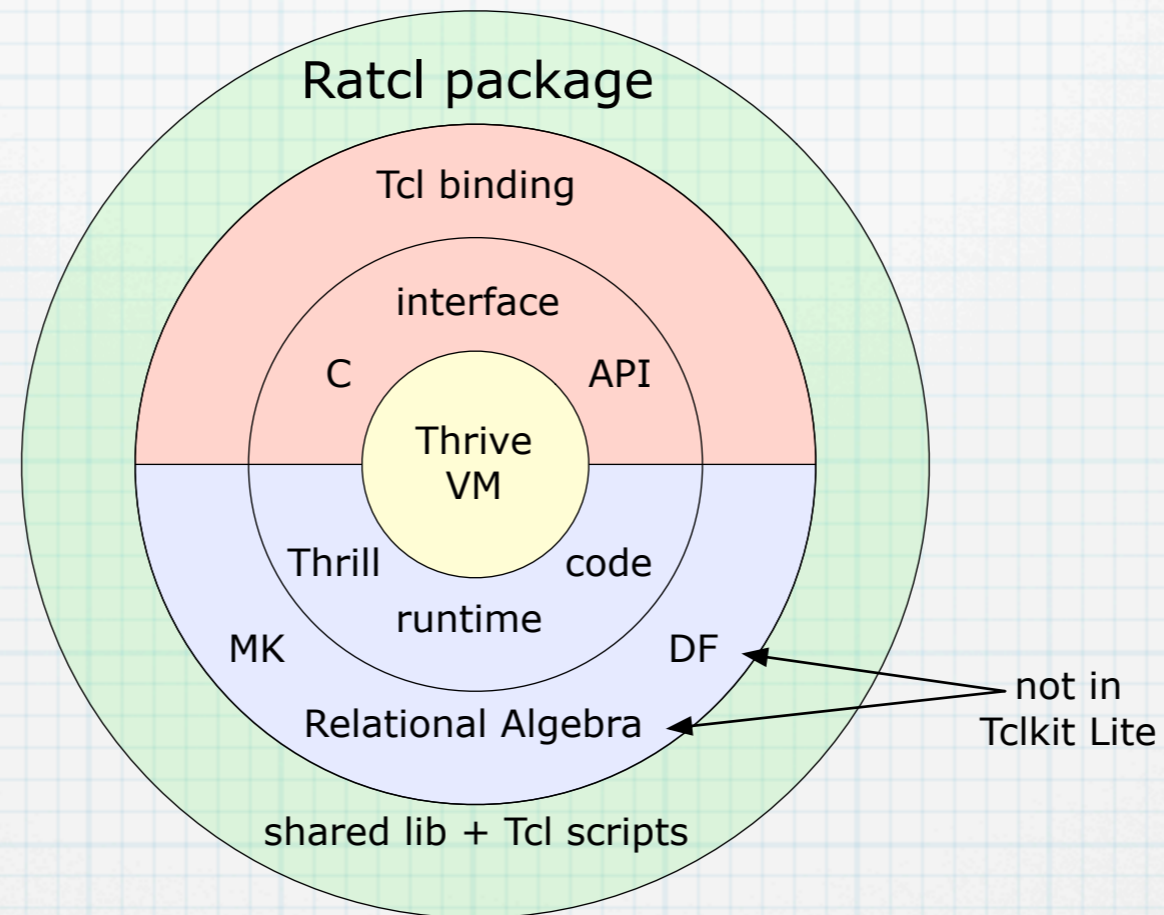
So what is Vlerq?

- * programming in a data-centric way
- * built-in persistence, forget load/save
- * use & manipulate set-wise, not loops
- * think in Tcl, code in Tcl, stay in Tcl
- * but what Vlerq REALLY aims to be:
 - * the Relational Spreadsheet!

Status - Oct 2005

- * core data engine is now ok - 5 rewrites
- * Telkit Lite - no more C++, new code, tiny
 - * performance similar: it's interpreted!
- * Ratcl 1.0 ready end 2005 - *kit < 50 Kb
 - * r/w, no commit, no dataflow, not fast
- * all MIT-licensed OSS @ www.vlerq.org

Architecture



MK = Metakit compatible persistent storage

DF = Dataflow logic & change propagation

Expectations

- * Mine are very high, unfortunately!
- * performance, flexibility, ease-of-use
- * speed: outperform Tcl lists/arrays
- * flexible: relational data & XML data
- * ease-of-use: Ratcl fades into Tcl, like Tk
- * dataflow: data-aware widget catalyst?

Lessons learned...

- * Metakit's file format is very effective
- * being different pays off: column-wise
- * the secret of Ratcl and MK is vectors
- * Vlerq is really about finding simplicity
- * it's great to have Vlerq funded by Eolas

Building Telkit Lite

- * `wget http://www.equi4.com/pub/tk/tars/genkit`
- * `telsh genkit A` « or telkit
- * `telsh genkit B tel` « or telkit
- * `sh genkit B lite` « - Mk4tcl, + Thrive
- * `sh genkit D lite` « good enough to try it
- * `sh genkit E lite` « needs a full telkit (!)