

JazzScheme: Evolution of a Lisp-Based Development System

Guillaume Cartier

Louis-Julien Guillemette

SFP 2010

Outline

- What is JazzScheme?
- Why JazzScheme?
- Some applications
- History
 - Port from C++ to Gambit
- Jedi IDE
- Future work

Why JazzScheme?

- Good question!

Roots

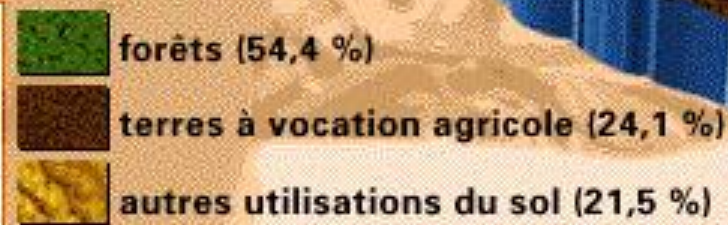
- Little Lisper
- Lisp machines
 - Complete programming environment
- Common Lisp
 - Macintosh Common Lisp
- Scheme
- Prisme (1990) (screenshots)
 - Highly graphical applications for real-life clients
 - Full access to the source code of the system

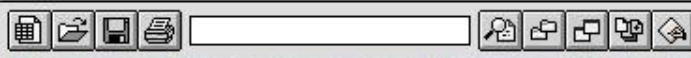


Pérou



Utilisation du sol





Add

Vnd Bhy Usr

OK ICM

☒ ○ ▾

☒ ☒ ☒

T: T

☒ ☒ ☒

Classes

Mode

☒ ☒ ☒ ☒ ☒ ☒

Tools

☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

Colors

Children

Esquisse -> inde

- fond
- amerique-sud
- do-quit
- amerique-nord
- afrique
- asie
- nil
- zone-active
- pays
 - drapeaux
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - nil
 - branch
 - do-quit
 - menu-principal
 - bande-titre

Properties

Esquisse : class Esquisse

- active-proc? True
- actor :parent
- auto-position? False
- auto-size? False
- auto-title? False
- branch
- branch-element ○
- change-on-tab? True
- color White-Color
- color-fore Black-Color
- content ○
- cursor :arrow
- data-base ○
- data-element ○
- data-links ○
- data-server :parent
- documents Empty
- drag-axis :unconstrained
- drag-constrained? False
- drag-constraint :desktop
- draggable? False
- draw-on-drag? False
- effect :copy
- effect-step ○
- effect-wait ○
- erase? True
- filter
- focus-hiliter
- focus-hiliter? False
- follow-parent? Empty
- font Font-System
- frame? False
- grow-icon? False
- handles-keys? False
- handles-mouse? True
- hide-on-close? False
- hilite-focus? False
- init-page
- locked-to-page? False
- menu Empty
- menu-model obj-menu-popup
- menu-parent Root-Menu
- menu-position 1 1
- mouse-copyable? False
- moveable? False
- next-effect ○
- offscreen? True
- opaque? False
- page-title
- pathname ○
- position :centered
- propagate-actor? False

Pages

obj-page

- intro-mp
- menu-principal
- commun
 - intro-dr
 - documents-reference
 - concepts
 - durabilite-sociale
 - projets
 - les-projets
 - general
 - choix-pays
 - asie
 - indonesie
 - inde
 - pakistan
 - nepal
 - thailande
 - bangladesh
 - afrique
 - amerique-nord
 - amerique-sud

Pages

obj-page

- intro-mp
- menu-principal
- commun
 - intro-dr
 - documents-reference
 - concepts
 - durabilite-sociale
 - projets
 - les-projets
 - general
 - choix-pays
 - asie
 - indonesie
 - inde
 - pakistan
 - nepal
 - thailande
 - bangladesh
 - afrique
 - amerique-nord
 - amerique-sud

Choix du pays

Esquisse d'un pays

Inde

Asie

Afrique

Amérique du Nord

Amérique du Sud

Quitter

What is JazzScheme?

Development system based on Scheme and Gambit

- Module system
- Hygienic macros
- Object-oriented programming
- Optional typing
- Cross-platform UI based on Cairo
- Binaries for Mac OS X, Windows and Linux
- Lisp-based IDE

Why (contd.)

- Build commercial software in Lisp
- Promote Lisp
 - Not by trying to convince people of its advantages
 - By creating a Lisp-based development system to write complex applications that would have been extremely difficult to develop using main stream languages (time, cost, feasibility, ...)

Requirements

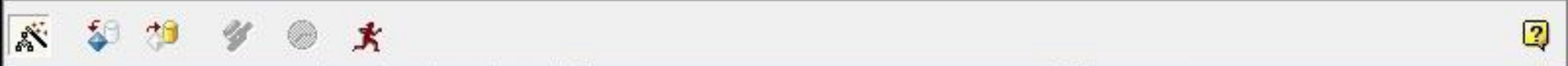
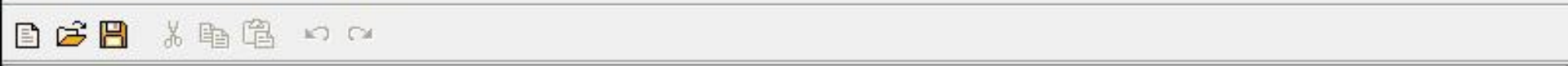
- Year 1998
- Large-scale enterprise development support
- Open-source
- Built entirely in its own language

Built entirely in Lisp

- Rapid development cycle
 - High-level language and tools
 - Only one language to learn and master
 - Fast evolution
- Live by your word
 - New features & optimisations
 - Constant testing
 - Tribute to Lisp
- Openness to the community

Some applications

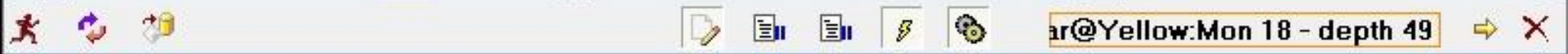
- MetaModeler
 - Database modeling
- Scheduler (screenshot)
 - Automated rule-based scheduler for hospitals
- Uranos
 - Enterprise Resource Planning (ERP)
- Jedi
 - Lisp-based IDE all written in Jazz



User	Min	#0	Max	Fail
Al-Ahmadi, Turki	75	60	90	
Al-Backer, Nouf	0	0	60	
Al-Fares, Ahmed	90	45	90	

Rule
Ward-sr
SER-sr
9C

Attribute
Surgery
Weekend
Night



Shift	Period	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
PICU-9D	17:00 - 08:00	MS	RAS	1616	MS	RAS	1616	MS	VH	RAS	MS	VH	RAS	2839	VH	2740	2988	2740	2740	2800
NICU-9C	17:00 - 08:00	TAA	9C	IA	1543	ZAI	TAA	IA	9C	TAA	9C	9C	ZAI	TAA	IA	ZAI	9C	2884	3247	3300
RVH/JGH	17:00 - 08:00	1247								1298		1276				1273	1276			
Ward sr	08:00 - 20:00	WAM	PL	WAM	PL	WAM	1616	WAM	PL	WAM	PL	WAM	PL	1564	PL	WAM	PL	WAM	PL	WAM
NF	20:00 - 08:00	SAZ	SAZ	SAZ	1799	SAZ	SAZ	1616	RF	1833	RF	RF	RF	RF	1564	SAZ	SAZ	SAZ	1819	SAZ
Blue	17:00 - 08:00	1715	1782	AR	KH	EST	AR	KH	1588	EST	KH	1564	AR	EST	KH	1561	EST	1561	1585	AR
Yellow	17:00 - 08:00	SS	AAF	OA	SS	AAF	1276	SS	OA	3027	2970	CT	OA	AAF	1276	3585	3475	3475	CT	3475
ER senior Day	08:00 - 17:00	1669	1669	1744	1748	1748	1748	1744	1748	1767	1671	1767	1771	1771	1767	1744	1744	1670	1679	1670
ER senior Evening	17:00 - 00:00	1669	1744	1744	1748	1744	1744	1744	1744	1679	1767	1767	1771	1767	1767	1744	1744	1679	1679	1680
ER senior Night	23:00 - 08:00	MW	ZAS	HV	MAG	MAG	NF	NF	MAG	JT	ZAS	ZAS	MW	HV	HV			JT	MW	
FR senior Night (float)	23:00 - 08:00															1561			AR	1700

Variable	Value
dirty?	#f
attribute-weights	<unbound>
rank	2
decisions	{Vector {@Scheduler-Decision Al-Backer
rules	{{@Scheduler-Contract-Rule #x6F839

Name	Result	Value

Birth of “classic” Jazz

- Year 1998
- C++-based

Birth of JazzScheme

- Year 2006
- Meeting Marc Feeley
- Jazz becomes open-source JazzScheme
- Scheme-based
 - (Chicken, PLT, Bigloo but not Gambit!)
- Port to Gambit

Why Gambit?

- Lightweight, high-quality Scheme implementation
- Conformance (R5RS and IEEE Scheme standards)
- Portability
- Performance
- Reliability
- Debugging
- Rich API
 - C foreign-function interface
 - Lightweight thread system that can support millions of concurrent threads
 - Networking
 - Unicode support

Port to Gambit

- Scheme was just too great!
 - Jazz becomes a radically different language
- We ended up having to
 - Port the language from C++ to Gambit
 - Port the libraries to the new incompatible language
 - Port the UI code from Windows to X11 and Mac OS X
- Lisp's syntax saves the day
- Port
 - 200,000 lines of C++
 - 15,000 lines of Scheme

Optimisations

- First working version
 - 95x slower than C++-based Jazz
- Statprof
 - Statistical profiler used to identify all the hotspots
- The current version
 - Gambit-based kernel now faster than the old C++-based kernel

Class-of

- `class-of`
 - Edit definitions
 - Edit references
- `%%class-of`
 - Multi-scheme Jazz (Chicken, PLT, Bigloo)
- **Back / forward navigation**

The present

- Auphelia
 - Christian Perreault
 - QT vs Jazz
 - Enterprise Resource Planning (ERP)
- Team
- Continuous evolution of the JazzScheme platform
- Part-time collaborators
 - Marc Feeley
 - Alex Shinn

C4

- Project
- Remote debugger
 - Backtrace
 - Inspect variable
 - Dynamic redefinition

Jedi

- Code walker
 - Compile time highlight of errors
- Snapshots
 - Uses the same debugger infrastructure as the Jazz, Gambit and SWANK debuggers
- Inspector
 - Full support for Gambit data types
- Profiler
- View explorer
 - F8 / Edit Action Handler on Start Profile

Future work

- Deterministic profiler
- Code coverage
- Console-based REPL
- Stepper
- Designer
- SWANK debugger

Resources

- Website: www.jazzscheme.org
 - Documentation
 - Tutorials
 - FAQ
- Discussion Group
 - groups.google.com/group/jazzscheme
- IRC on Freenode
 - #jazzscheme
 - #gambit

Thank you!

Jazz as a macro

- Usual language development approach
 - Write a compiler that generates to a target language
 - Write the runtime support (GC, memory management, ...)
 - Write an interpreter (optional)
- As a macro
 - Reuse of all the work invested in Gambit
 - Written in a high-level language
 - Only a code walker needs to be developed
 - No performance penalty

Text

- Find definitions & references
- Dynamic redefinition
- Incremental search
- Search & replace with IrRegex
- Syntax highlighting
- Clipboard ring
- Mouse copy
- Emacs