**Fidji**

**Design notes and more**

*July 5, 2013*   
  
Miguel Calejo

TABLE OF CONTENTS

1 TODO IN THIS DOCUMENT 3

2 Introduction 3

2.1 History and acknowledgements 3

3 Architecture 4

4 Prolog listener 4

4.1 Bugs 4

4.2 Missing features 4

5 Editor window 4

5.1 Bugs 5

5.2 Missing features 5

6 Query window 5

6.1 Bugs 5

6.2 Missing features 5

7 Tabled Data 6

7.1 Bugs 6

7.2 Missing features 6

8 Graph windows 6

8.1 User notes 6

8.2 Bugs 7

8.3 Features 7

9 Project/workspace window 7

9.1 Bugs 7

9.2 Missing features 7

10 Other components 8

10.1 Forest logging (TBD) 8

10.2 SCC visualization ? (TBD) 8

10.3 Preferences and startup control 8

11 Infrastructure issues 9

11.1 XJ specific 9

11.2 InterProlog specific 9

11.3 Overall Fiji 11

# TODO IN THIS DOCUMENT

Strip/review: Flora, Sik

# Introduction

Fiji[[1]](#footnote-1) is an Integrated Development Environment for Flora and XSB Prolog, to which it adds a thin but powerful Java layer, providing syntax-sensitive editors, query control, tabled data browsing and more.

## History and acknowledgements

Fiji is my latest incursion into logic programming environment tools. I've been at it on and off since 1986, so for me this is a "complete the dots" story... It is being a gratifying exercise (specially between debugging sessions!), as I see how decades of accumulated ideas and techniques, plus the standing over many shoulders of giants... can shatter code and development metrics; the first "alpha" version of Fiji took just a couple of months to write, which is interesting since I'm a slow programmer and had significant distractions over that period.

In the "shoulders of giants department" I'd like to mention David S. Warren and Terry Swift, who both created XSB and cofounded XSB Inc and also gave me - with Rupert Hopkins- the environment to develop two iterations of XJ over a couple of years, a decade ago, which became (almost secret:-)) open source some years later. In addition to Rupert's critical support, David, Tanya Vidrevich, Harpreet Singh and others at XSB Inc. later added good pieces to XJ. And of course Java and Swing, which I continue to like more and more, especially to serve logic programming :-) Plus the best text API I've worked with, from the guys at Filesoft.

More recently, Benjamin Grosof and the Silk project gave me the chance to warm up to this; and later he and other co-founders at Coherent Knowledge Systems, namely Michael Kifer and Janine Bloomfield, tested and provided feedback in the context of "Fidji" - a preliminary proprietary IDE project of which the present Fiji is the infra structure.

Fiji's bridging nature (between **logic programming** and its **applications**) reflects onto its splash screen – which for the curious is *not* SF's Golden Gate... but the Lisbon bridge, as seen from the attic where I wrote most of it.

### External dependencies

Fiji benefited from my "intellectual warm up" in Silk, but has no dependencies on proprietary code nor Intellectual Property. It uses the following open source (or close) components:

- Java 6

- XSB

- InterProlog

- XJ, LGPL (gone offline as XSB revamped their site recently; it will return online - at XSB Inc. or elsewhere - after it stabilizes after getting the Fiji treatment)

- RsyntaxTextArea, modified BSD

- JUNG Graph Framework, BSD

# Architecture

Fiji is currently a Java app using InterProlog and a couple of xsb subprocesses. The GUI layer is implemented on Java Foundation Classes (Swing), complemented with two open source frameworks: a text editor library RSyntaxTextArea), and a Java+Prolog GUI generation component (XJ).

The Java side controls two XSB instances: the main one to load programs and execute queries, and an auxiliary for (source code editor) parsing and call graph representation.

In the future **other packaging alternative** are possible, such as a "MDI interface" (using XJDesktop) or custom embedding in other developers apps. Beyond the Java desktop, becoming a cloud IDE is a possibility (by developing a Javascript+Java(server-side) XJ); this being also probably the best road to tablets and smartphones.

Fiji components appear visually in the following screenshot:

*PICTURE TO APPEAR HERE*

# Prolog listener

The listener window exposes XSB's top level goal shell, including its tracer. It's a Java-side component, derived from the old InterProlog listener window, which in Fiji assumes a more important role as it acts (at least for now) as a center point for reaching other Fiji windows.

The listener is a "console-type" window, likely to be kept hidden from non-gurus, as it can break (or at least disturb) the GUI layer, which depends on calling Prolog itself often. The GUI layer disables itself partially when Prolog is unavailable to it, such as when a user query is running. So be nice to it: if you use the tracer, if possible don't use the mouse until you finish the trace and are back to the XSB shell (Fiji will later inject the notrace for you automatically).

## Bugs

\* REGEX not working effectively to simplify spurious XSB trivial goal success messages

\*\* This should work better than the current regex, but does not work:

static final String TRIVIAL\_FLORA\_OUTPUT = "(flora2 \\?- \\"+"nElapsed \\(CPU\\) time [0-9\\.]+ \\([0-9\\.]+\\) seconds\\"+"n\\"+"nYes\\"+"n\\"+"n)+";

...although it works on TextWrangler

\* sometimes loaded files do not appear in Open Loaded; requires pack()..??? or equals() funny business...?

\* Cmd-Q on Mac exits the application without asking for saving etc.; apparently some action other then the File menu's is bound to VK\_Q

## Missing features

* Prompt appearing (non editable) in bottom pane and not on output pane... cf. MK suggestion email June 18 2013

# Editor window

An editor window edits one Prolog file, providing functionality specific to the language; in particular, a Prolog parser runs in background (on a dedicated XSB engine) to provide in place warnings etc. Coloring for a couple of dozen languages (Java, HTML, etc.) is also bundled in courtesy of RSTA.

## Bugs

\* RTEXT opens big files

RSTA crashes, logging out the user!! Strange because RTextEditorPane doesn't seem that diferent..

Prolog files up to 8 Mb load on Mac, after that... Windows server error; buffering or delayijng the visual hierarchy does not change it! On Windows 7, up to 13 Mb opened fine

... we should tolerate parsing engine death, immediately null-ifying and relaunching it

## Missing features

New Prolog: fill with template header, or possibly other templates based on UI context (MK suggestion)

More clever indenting and unindenting, à la Emacs C mode; implies changing insertNewlineWithAutoIndent in RSTA (MK request)

\* Make word selection sensitive to tokens, namely variables and literals. cf. SelectWordAction implementation

\* steal features from RText: see constructor in RTextMenuBar

\* We need dynamic token sets to color according to just in time semantics; possibly second step on JFlex's tokens

\*\* User-defined in Prolog

\* More stuff

\*\* Review TextWrangler favorites other than "copy path"

### Dependent on project/workspace?

Comment tooltip

predicate/word completion

\*\*classify identifiers as to location... head, body?

# Query window

Where the user can pose queries, reasonably restrained by default, optionally stop or pause/continue them, get answers, and enquire the reason of undefined-ness.

It's a XJ (Prolog) specified window, without specific Java code.

## Bugs

Apparently after a large solution list is displayed, when it goes into history; (even with a second tiny query, with Execute) Java serialization blows up. Is it HistoryGoal, or excessive serialization of PrologAction?

## Missing features

\* Prolog justification

\* Snap shoot

\* Brat undefined could use third XJ list item type, possibly yellow and tip "..."

\* Incremental queries?

\*\* find all checkbox

\*\* More? button at end of answer list (third GT template)

\*\* Exception in list (fourth GT template)

\*\* G --> goal(G,.....): Iterator reference kept

\*\*\* initial query does hasNext,next

\*\*\* More? does hasNext,next

\*\*\* each next() call returns TermModel array

\*\*\* may or not succeed binding names to free vars

\* WHY as button in addition to menu

\*\* Uncomment stuff... and fix cosmetic issues and direct button event handling

\*\* dummy item at end "Why no more ?"

\* Use XJPrologTextArea?

\* after we stabilize the rest: provide \_leash(G) for use in listener

Up/down arrows navigating in history; adapt solution from prologInput field in ListenerWindow (MK request)

# Tabled Data

A browser of the tabled (XSB) subgoals and answers, with 3 layers: predicates, subgoals (red if failed, green otherwise) and answers (green!). Incremental tables (predicates) in orange.

## Bugs

## Missing features

* Add filters – either with history or let's just use multiple window instances
* start using SFs. Then quick access to get\_callsto\_number (there's no other way...), get\_answer\_number
* Hottest calls: find 5; after this, check if table\_dump is necessary
* Add pretty printing to nodes
* Offer to show suffix @main - preference??
* Optimize nodeIDs: use table "pointers"
* See incomplete tables in table browser
* Refresh tables (selected and all)

\*\* how to complete: force\_answer\_true/false? early\_complete\_on\_nth? EARLY\_COMPLETE? Terry will follow-up

# Graph windows

Based on a new XJ component using JUNG.

## User notes

When using "View results as graph", beware that answers with free variables may lead to more edges than expected; for example ?- ?X=1;?Y=2 originates a graph with 5 edges.

## Bugs

check expansion of neighbors too, in collapseNodes()

## Features

\* Use the 3 extra Forest/tree layouts

\*\* Refactor LazyGraphModel into LazyGraphModel + GraphBackend (delegate?)

\*\* Add LazyForestModel extends DelegateForest, also using GraphBackend delegate

\*\* Graph interface

\* Memory optimization

\*\* Factor edgeRelation, and differ fecthing of its attributes as well as of nodeRelation

\* Reuse Pair and Node objects, too many remain created afterwards

\* different node shapes

\* Copy JPEG or PNG

\* Persist layout

\* icons on edges: use HTML, cf ImageEdgeLabelDemo

\* Satellite view, with finder, using JUNG predicates

\* See PersistentLayoutDemo for persistence preferred layouts of important graphs

\*\* Create a PersistentLayoutImpl wrapping our layout, and provide a temp file for persistence

\*\* Improve initializeLocation to tolerate mismatches in vertices

\* operations for the whole graph are unlike XJ trees and lists in that they have no terms(...,...) lambdas: they force the explicit use of getSelectedNodeIDs, and they do not disable themselves on empty selection

# Project/workspace window

Shows a list of Prolog files and their load/add state with navigation to their editors. Files can be dragged from the user desktop.

## Bugs

## Missing features

* Add projects to Windows submenu
* Save As (for now users can simply copy the file and rename it...)
* Prolog loads always "succeed", lacking control
* Minimize/close all
* Factor file paths, offer to consolidate into single directory?

# Other components

## Forest logging (TBD)

Still unclear whether Fiji will show forest log material per se; more likely to support the implementation of some justifications or to appear in restricted form closer to the query window.

## SCC visualization ? (TBD)

Still unclear (to me!) whether Fiji should show SCC dependency graphs, or just an actionable synthesis of it.

## Preferences and startup control

Being a Java application dependent on an independent language system – XSB Prolog – Fiji tries nevertheless to be as "installation-free" as possible, so users can dive into using it as quickly as possible:

* The simplest procedure is to (1) download a self-contained zip file (which includes Fiji and XSB), (2) extract its contents and (3) doubleclick the fiji.jar file. This is also the safest approach, as that XSB will typically be the latest and compatible with Fiji. *How does it work: Fiji follows a simple naming convention – if it finds FijiXSB in its directory, it assumes it is the XSB to use*.
* For users wishing to use a preinstalled XSB, (1) download fiji.jar, (2) doubleclick it and (3) when asked indicate the location of the existing XSB executable.

If something goes wrong during startup, Fiji should detect the problem and refuse to run, presenting an error dialog; or it may even draft an email bug report. *How does it work: in addition to testing several system preconditions, Fiji has an autonomous startup timeout detector, which kicks in after about 20 seconds*.

If something goes wrong later, try to use Fiji's **bug reporter**. It's in the listener window, Tools/BUG REPORT.

### Technical details

Fiji stores invisible preferences and log files in its directory, so it needs write permission there.

When ran directly from the user's desktop (e.g. doubleclick from Finder or Explorer), all its Java out and err output will be redirected to .Fiji.log.

Fiji can also be run with more visible Java output, using the OS shell with:

java -classpath Fiji.jar com.declarativa.fiji.FijiSubprocessEngineWindow Path\_to\_xsb\_executable (but if you do this, make sure that the xsb executable you indicate is compatible with the Flora directory you'll pick when asked by Fidji).

### Bugs

On Unix XSB must be installed in a directory without spaces in its path. See related InterProlog bug.

### Missing features

Nicer startup: use Java 6's SplashScreen?

Possible startup wrapping: http://launch4j.sourceforge.net/ ?

website distributing Fiji should also advise on proper zip/installer according to browser OS

# Infrastructure issues

## XJ specific

Fiji uses a slightly improved XJ, including some important improvements such as lazy (level) expansion of lazy trees (:-)), so it doesn't choke on trees with large breath factors; minimal Flora support; etc.

### Bugs

Lazy lists can not have a nonatomic goal, otherwise sorting will crash because of parsort/4 domain error

### Missing features

\* Factorize icon searching code in XJButton.java:98

\* allow Java-side operations: if goal=javaMessage(Target,Message), Message args must be myGUI vars or the root...

\*\* Useful to specify UI in Prolog and wire to Java event handlers

\* Dynamic tips in lists and trees require a convention to pass the tip...e.g. iamtip property, indicating node providing data

\*\* For problems panel (file)

\*\* For undefined solutions (reason)

\* XJ trees, lists, graphs with node-dependent property values, to avoid template explosion

\*\* approach

\*\*\* extra in node term relations to obtain adhoc list of properties

\*\*\* fabricate extra prop list in node rendering goal?

\*\*\* access gt(Data,....) values from template properties?

\*\*\*\* VariableNode cold have a value hanging...?

\*\* difficulty

\*\*\* renderers are generated at GUI setup time...

\*\*\* ...so makeGUI har already ran...

\*\*\* would it be ok to run makeGUI EVERY time?? ...

\*\*\* ...perhaps in a "refreshProperties" mode only?

\* XJTree with eager model for better term display?

\*\* NodeType(Node,Children)

\*\* Node types: Hilog head, ...

\* xjJavaMessage(...), which if receiving a XJMessageException(Message) will call xj\_failError(Message)

## InterProlog specific

Fiji uses an InterProlog slightly more functional than the one in [www.declarativa.com/interprolog](http://www.declarativa.com/interprolog). Namely it has basic support for the Flora2 language, and better performance.

### Bugs

Synchronizing setResult in GoalToExecute causes deadlock on Prolog death (SubprocessEngine.java:257), hence causing the exception to be NOT thrown. Should use a finer grained semaphor.

Consider using a separate xsb starter, to avoid using the "broken process" API

### Features (including performance)

\* Save temp Strings in IFTM inflateFromStream; remember to doubleQuotes on the stream

\* ObjectInputStream.readUTF: intern!

\*\* Use MyObjectInputStream in SubprocessEngine.receiveObject and AbstractNativeEngine.callback

\* FloraSubprocessEngine

\*\* Flora query with undefined answers

\*\*\* Simpler approach?, lets look at terms; use perhaps flora\_decode\_oid\_as\_atom/2

\*\* HiLogTermModel ??

\*\*\* Knowledge about Primitive data/symbol types?

\*\* Add FloraSubprocessEngine test suite demoing Prolog and Flora querying

??? Can we make a specific class to obtain this performance??

\* TermModel children[] .... --> Object[], elements are either basic types wrappers or TermModel

\*\* E.g., if a child i is a leaf, we put its node in children[i]

\*\* getChild... should wrap in TermModel... those in the know access getRawChild:-)

\*\* Expected edges profit...13-->2 TermModel instances per Terry graph edge!

\* stream of object: resets handles, both in Java and Prolog

\* mmap later

\*\* this can also avoid sockets if we need less latency:-)

\*\* http://cdn-static.zdnet.com/i/story/60/40/001327/jfsoptimized.png

\*\* sched\_yield?

\* MyStreamTokenizer should be better with ints...might want to grab the older version (r210!) for ints

\* We need to timeout startup and InterProlog in general, to harakiri if things fail...?

\* Restrained deterministicGoal, cf. Silk recipe

\* sync SVN Unfuddle with local repository (Dropboxed); or backup

\* redo documentation as wiki pages (where?)

\*\* merge /Users/mc/Dropbox/interprolog212x/QUE\_É\_ISTO.txt

\*\* document how to build xsb for gdb debugging, provide script

\*\* Add to FAQ

\*\*\* Huge Java objects with circular references (e.g. JFrames) hang the grammar; use an object reference instead

\*\*\* "how do I return an integer/float/whatever", ProductEntry/Catalog example (XJ 2000)

\*\* review doc based on CHANGES refs

\* Change license references, cf. email Sergio Castro Mar 20 2013

\* release 3.0, emailing to all interested parties plus APPIA etc

Later:

\* ipObjectSpec(Label,......) allows javaMessage(....,Label(Args))

\* Terry, May 1 telecon: at some point we should have an interprolog test suite to make sure that changes to XSB dont break anything

\* add interprolog212x IP native speedups, question use of atoms

\* commit flexible change to XSB (extra interprolog\_callback argument)

\*\* make IP smart about older XSB

\* variant\_get\_residual/2 only for tabled preds, needs to be used adhoc

\*\* future field: TermModel lastSolutionWithResidual To avoid eagerly pushing residual to Java

\*\* TermModel[] lastSolutionResidual()

## Overall Fiji

### Bugs

Review all dg exception handling

\* Identify dg calls and XJ actions in the AWT thread. Apply an ImpatientBR?

\* make fidji.P a module? may require making some usermod: calls explicit?

\* System-wide, we might use addAWTEventListener to detect (in conjunction with engine load) hangups in general

Refuse to run on older systems (namely on XP because of gpp)

-921\trailer$eq=none+inh=flogic+tbl=nonincremental+nonincremental+cus=none.P]

\* Insert headers with license and $ID stuff

### Missing features

\* make UI partially "user"-scriptable...??

1. Fidji is written in the French manner, with a D in the middle; so we keep an European ring to it:-) [↑](#footnote-ref-1)