plwm

An X11 window manager written in Prolog

Barnabás Zahorán

<zahoranb@proton.me>

The project

Dynamic tiling window manager for X11 (like dwm)

v0.5 is stable for daily use

Written in SWI-Prolog

Porting underway to Scryer Prolog

github.com/Seeker04/plwm

Declarative window management

Started as a proof of concept

Grew to be a working solution

Novelty: first WM written in Prolog

(to my best knowledge)

Initial concerns

Performance?

Scalability of code?

Interoperability?

Tooling?

Determinism?

Is Prolog **practical** for tasks far from its primary uses?

Configuration

A breeze in Prolog thanks to its declarative nature

We simply consult a source file

Can be reloaded at any time

Dynamic predicates to change settings on the fly

Settings are simple facts usually, but can contain computation

We can define custom predicates to execute later

Features

Fast and lightweight

12 tiling layouts, floating and fullscreen windows

Dynamic workspaces: create, rename, reindex, delete them

Multi-monitor support

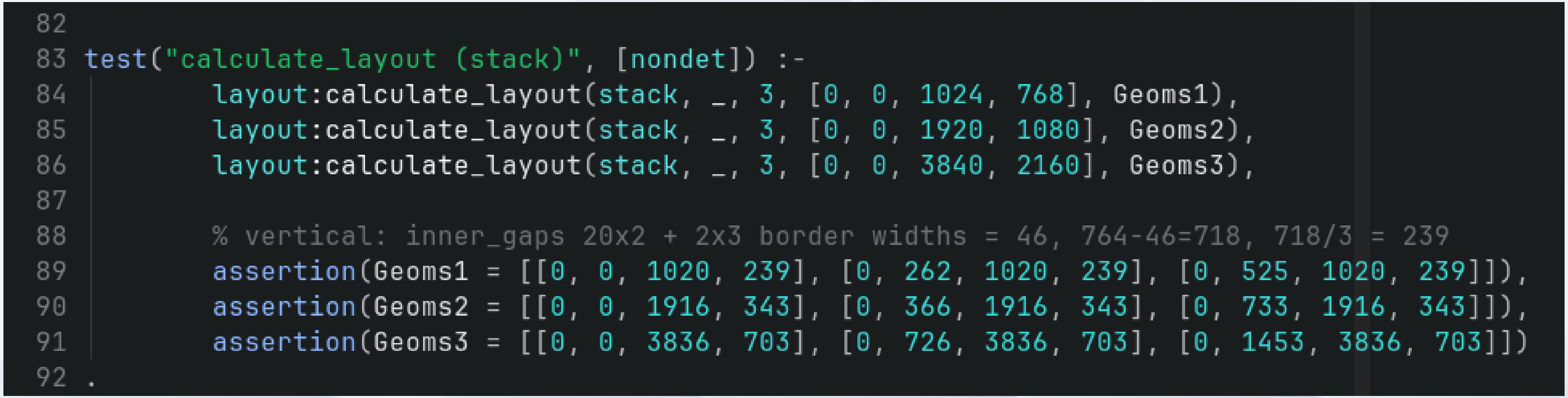
Simple animations

Hooks

Command menus and FIFO → custom user logic and scriptability

Building relations for a WM

```
calculate_layout(Layout, Mon, WinCnt, Bounds, Geoms)
```



```
test("calculate_layout (stack)", [nondet]) :-
        layout:calculate_layout(stack, _, 3, [0, 0, 1024, 768], Geoms1),
        layout:calculate_layout(stack, _, 3, [0, 0, 1920, 1080], Geoms2),
        layout:calculate_layout(stack, _, 3, [0, 0, 3840, 2160], Geoms3),

        % vertical: inner_gaps 20x2 + 2x3 border widths = 46, 764-46=718, 718/3 = 239
        assertion(Geoms1 = [[0, 0, 1020, 239], [0, 262, 1020, 239], [0, 525, 1020, 239]]),
        assertion(Geoms2 = [[0, 0, 1916, 343], [0, 366, 1916, 343], [0, 733, 1916, 343]]),
        assertion(Geoms3 = [[0, 0, 3836, 703], [0, 726, 3836, 703], [0, 1453, 3836, 703]])
.
```

Testing

In Prolog the code tests itself

Assertion: we usually make all arguments ground

Mocking: easy with dynamic predicates/globals

PlUnit: not essential, but has some nice QoL like

- coloured output
- result aggregation
- setup & cleanup steps for each test

Quirks

Prolog is crazy good at prototyping

simple syntax and semantics, REPL, GC, higher-order predicates
→ fast to develop and iterate

Weak type system

it actually wills us towards simpler solutions
and not overengineered ones

Debugging is a little cumbersome

though it's more of a WM nuance than a Prolog one

Module system

is a little all over the place...

Porting to Scryer

Credit: **Bennet Bleßmann** `@Skgland`

Fork: `github.com/Skgland/plwm/`

We have an ongoing draft pull request

Status:

- **We can build for both SWI and Scryer**
- **SWI retains functionality, but needs some testing**
- **Core window management works in the Scryer version! Testing is underway and issues are being fixed**

Porting to Scryer

Tasks and pitfalls

Missing predicates present only in SWI

Non-ISO compliant syntax in some existing code (e.g. ambiguity)

Stricter argument types in Scryer

FFI to C is conceptually different:

- **SWI: we write bindings in C using SWI-Prolog.h**
- **Scryer: we bind from Prolog with library(ffi) using libffi**

Module system differences…

Porting to Scryer

Module system

Many predicates in SWI are builtins which we must import

Also, a lot of modules are auto-loaded in SWI

SWI predicates are looked up in the current module and its import module chain (usually current → `user` → `system`)

Scryer only looks up predicates in the current modules and in the builtins, so explicit `user:` qualifications are needed

No conditional declarations in Scryer, i.e. we can’t simply:

```
:- is_scryer -> use_module(library(iso_ext)) ; true.
```

Had to jump through some hoops instead (especially for meta predicates)

Porting to Scryer

Scryer functionality was developed and merged specifically for this porting:

- library(process) (#3009)
- support for non-fixed-width int types for ffi (#3111)

`goal_expansion/2`

Super handy tool for prototyping and providing functionality for missing SWI predicates

Porting to Scryer

TODOs:

Missing multi-threading (for animations and FIFO)

Signal handling (graceful shutdown on SIGTERM)

Fix known issues (hooks, menus)

More testing...

Back to our concerns...

Performance? – Negligible, plwm feels as snappy as dwm

Scalability of code? – Scaled nicely to 5000 LOC

Interoperability? – Binding to C is easy with FFI in both SWI and Scryer

Tooling? – There’s room for improvements:
LSP, DAP, linting, formatting, package manager (there are initiatives)

Determinism? – Non-determinism was a handy tool and, with well structured code, didn’t get in the way (we can use `once/1` where needed)

Is Prolog practical for tasks far from its primary uses? – Yes, in my opinion

Sponsoring this presentation

ASTRON
business upgraded

Trusted Technology Partner for Mission-Critical Systems

Real-World Impact, Global Reach

About Astron

Mission

We’re a trusted IT partner, creating **mission-critical**, **custom software solutions** that help large organizations succeed worldwide.

Business areas

- Energy sector - Electric Power and Natural Gas Industry
- Transportation and Mobility
- Mobile App Development

Key Facts

25+ years
of experience

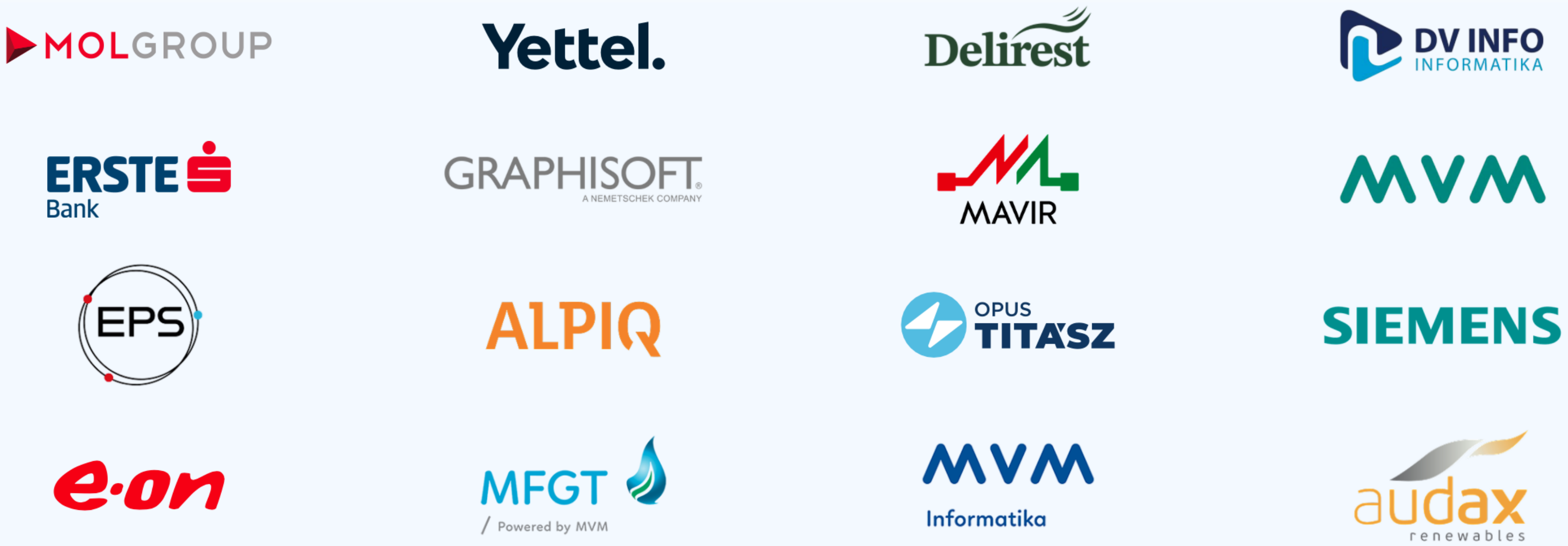
8,25 M Euro
revenue

1300+
projects delivered

80+
skilled professionals

ASTRON
business upgraded

Trusted by Industry Leaders



> "Trustful, reliable partner with state-of-the-art knowledge."
> Michael Ohrstedt
> Head of TMS Mainline Systems, Siemens

ASTRON
business upgraded

Information Security & Compliance at the Core

ISO 9001
Quality Management

ISO 27001
Information Security Management

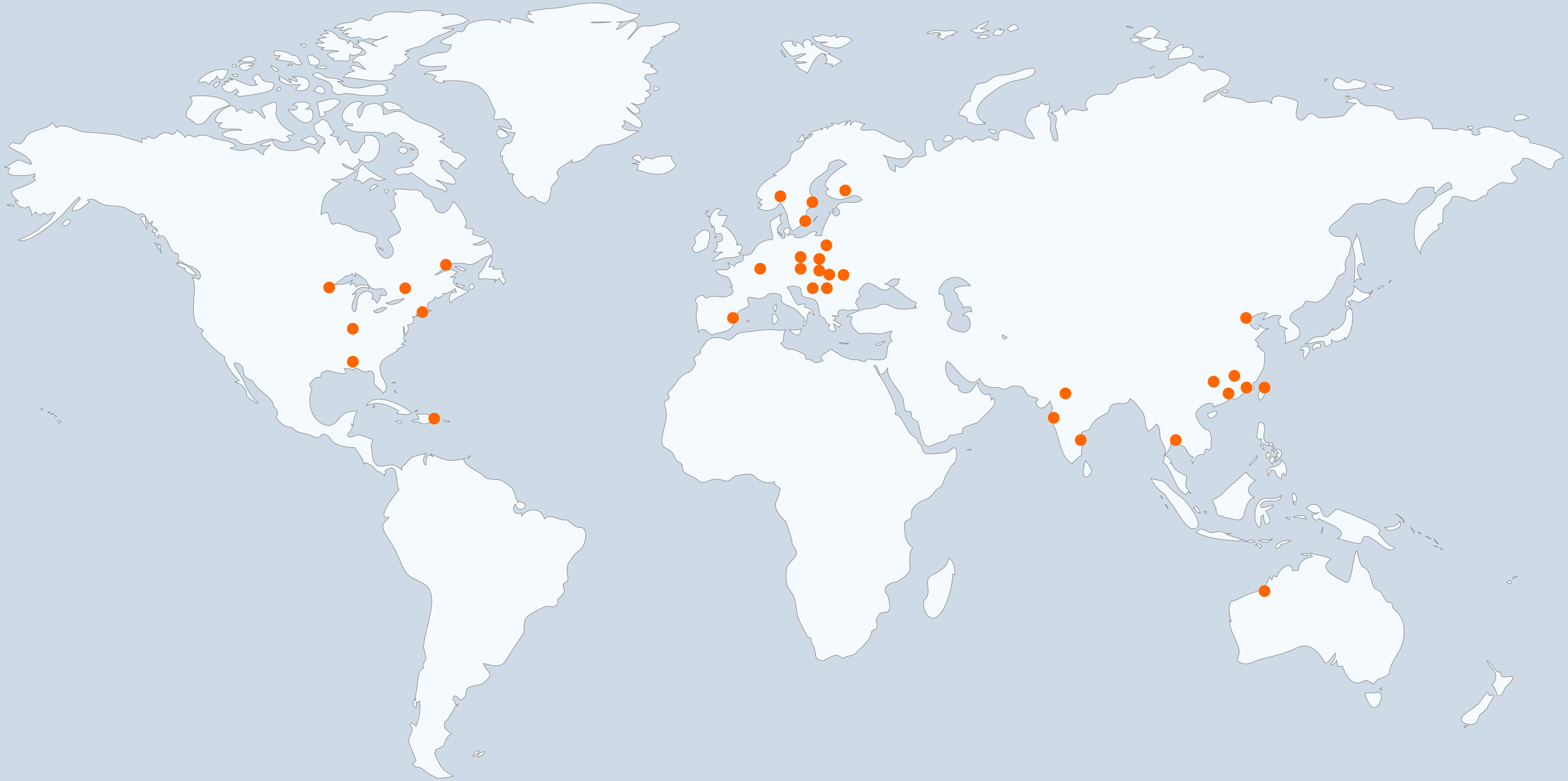
ISO 20000
IT Service Management

CENELEC
EUROPEAN COMMITTEE
FOR ELECTROTECHNICAL STANDARDIZATION

Ensuring reliability, safety, and trust in every solution we deliver.

ASTRON
business upgraded

Global Reach



ASTRON
business upgraded

Summary

- ✓ International project experience
- ✓ 25+ years of field experience
- ✓ Top-trained software engineers
- ✓ Long-term partnerships with global leaders such as Siemens, MOL, and Yettel

> “Astron Informatics is a stable and trusted partner, providing state-of-the-art IT solutions and deep industrial expertise.”
>
> Ákos Kriston,
>
> CEO, Hungarian Gas Storage Ltd.

ASTRON
business upgraded

Feel free to reach out!

ASTRON

business upgraded

ASTRON INFORMATICS Ltd.
H-1119 Budapest, Budafoki út 56.
www.astron.hu