Introduction to Functional Programming in *OCaml*

Roberto Di Cosmo, Yann Régis-Gianas, Ralf Treinen

Week 0 - Sequence 3:

Why *OCaml* : voices from the user base

Who uses the *OCaml* language?

Teaching

- France: University Paris Diderot, Pierre et Marie Curie, Paris Saclay, Rennes, ...
- Europe: University of Pisa, Bologna, Birmingham, Cambridge, Aarhus, Innsbruck, Wroclaw, ...
- United States: Cornell University, Harvard, MIT, Pennsylvania, ...
- and many others...

Who uses the *OCaml* language?

Advanced Research Projects

- Coq proof assistant (ACM Software System Award 2014)
- Astrée static analyzer (verifies Airbus A380's code)
- Frama-C platform (analysis of C code)
- Ocsigen (advanced web application framework)
- Alt-Ergo (advanced SMT solver)
- Mirage OS (unikernel)
- Flow / Hack (PHP/Javascript type checkers)
- and many others ....

Who uses the *OCaml* language?

Industry

- Bloomberg, *finance*
- Citrix, *virtualisation, cloud*
- Dassault, *aerospace*
- Facebook
- JaneStreet Capital, *finance*
- LexiFi, *finance*
- Microsoft
- RedHat
- ...

Let's hear what all these people say...

Ensuring safety of critical embedded code: Astrée

`Astrée` is the static analyzer used by Airbus to prove the A380's command and control code bug free.

> *A type-safe functional language was the natural choice to implement the Astrée analyzer.* OCaml*'s robust design supported a scalable development process, from research to industry, and we appreciated its high performance native code compiler.*
>
> *Antoine Miné, Researcher at CNRS & ENS (2015)*

Mechanized Proofs: The Coq Proof Assistant

The Coq proof assistant is a formal proof management system.

> *Amongst all the great features of* OCaml, *pattern matching is crucial for Coq: without it, implementing complex symbolic computations would be a nightmare!*
>
> *The Coq development team (2015)*

Cybersecurity: TrustInSoft

`TrustInSoft` provides innovative software safety and security solutions.

> OCaml *generates code that's very efficient compared to other languages with similar expressivity. Expressivity is needed when developing sophisticated static analyzers. Efficiency is necessary when working at the frontier of what is possible at all on today's computers. Static typing saves clock cycles at execution time and, more importantly, human time during development.*
>
> *Pascal Cuoq, TrustInSoft (2015)*

Next generation web applications: Ocsigen

The `Ocsigen` project allows to write amazing web applications.

> OCaml's type system allows Ocsigen to check statically advanced properties of a Web application, like ensuring that a program will never generate invalid HTML pages, or that a form has the expected fields.
>
> The advantages of this powerful type system become obvious when refactoring a large project: the compiler points out every piece of code that needs to be modified, saving days of testing and debugging.
>
> Vincent Balat, creator of Ocsigen (2015)

Development tools: OCamlPro

`OCamlPro` specialises in *OCaml* development.

> *I have tried many programming languages, but none of them could compete with* OCaml. *In* OCaml, *you just define the type of your data, and the compiler will gently drive you towards your destination, at highspeed on a highway. It's just fascinating!*
>
> *Fabrice Le Fessant, OCamlPro (2015)*

Cryptography: Cryptosense

`Cryptosense` develops vulnerability assessment software for cryptography.

> *We see* OCaml *as a strategic advantage. It helps us to rapidly produce high-quality readable, reusable code, which is essential for a start-up.*
>
> *Graham Steel, Cryptosense (2015)*

Finance: LexiFi

`LexiFi` creates innovative software for managing complex financial products, combining advanced symbolic manipulations and numeric computations.

> *Safety, readability, expressivity and great performance are often cited as key benefits of* OCaml. *We also value the portability of the system, as our products are deployed on Unix, Windows and in the web browser.*
> *Parts of our codebase which were historically written in C, C# or Javascript are now in* OCaml. *As one of the earliest industrial adopters of* OCaml, *we are delighted to observe the growing interest and activity around* OCaml *in the last years.*
>
> *Alain Frisch, LexiFi (2015)*

Operating Systems: Mirage

Citrix and Cambridge University are now developing Mirage OS, a baremetal exokernel for Xen fully written in *OCaml* !

> OCaml*'s combination of static type safety and fast native code compilation has been essential to our MirageOS project, which rebuilds operating system components (including TCP/IP and device drivers) in a safe, modular and flexible style.*
>
> *Anil Madhavapeddy, Cambridge University (2015)*

Finance: JaneStreet

`JaneStreet` uses *OCaml* for building financial trading tools
that handle 10 Billions dollars per day

> *Our experience with* OCaml *on the research side convinced us that we could build smaller, simpler, easier-to-understand systems in* OCaml *than we could in languages such as Java or C#. For an organization that valued readability, this was a huge win...*
>
> *There is, a surprisingly wide swath of bugs against which the type system is effective, including many bugs that are quite hard to get at through testing.*

Yaron Minsky.
OCaml for the masses.
Communications of the ACM, September 2011

Virtualisation and cloud computing: Citrix, Xen

`Xen` is the *hypervisor* that powers millions of virtual machines in the `cloud`.
Its management tools are written in *OCaml*.

> OCaml *has brought significant productivity and efficiency benefits to the project.* OCaml *has enabled our engineers to be more productive than they would have been had they adopted any of the mainstream languages.*
>
> *Richard Sharp, Citrix*

To sum up

There is a wide variety of users of the *OCaml* language

They value unanimously:

safety
: from strong static typing
and pattern matching

efficiency
: a high performance compiler

expressiveness
: combination of a functional language
with type inference and polymorphism

We'll see a quick selection of examples to get a taste of all this.