Generative XPath

Uniting theory and practice

Oleg Parashchenko, Saint-Petersburg State University, Russia

Practical side. A complete XPath 1.0 implementation which can be adapted to different hierarchical memory structures and different programming languages.

Useful:

- For compilers and interpreters
- For text processors
- As a replacement of any sufficiently complicated tree navigation library

Theoretical side. No recursion construction is used except for list comprehension and morphisms.

Useful for testing algebra of programming on a real-life system.

Deploying

To add XPath support to an application, the developer needs to:

- Find or create a suitable VM implementation
- Implement the customization layer

Sample project. Evgeni Milenin's Master Diploma (SPbSU, 2007): "XPath over S-expressions". An unofficial title was: "XLinq for Lisp".

Links

My research wiki with a lot of links: `http://xmlhack.ru/protva/xquery/`

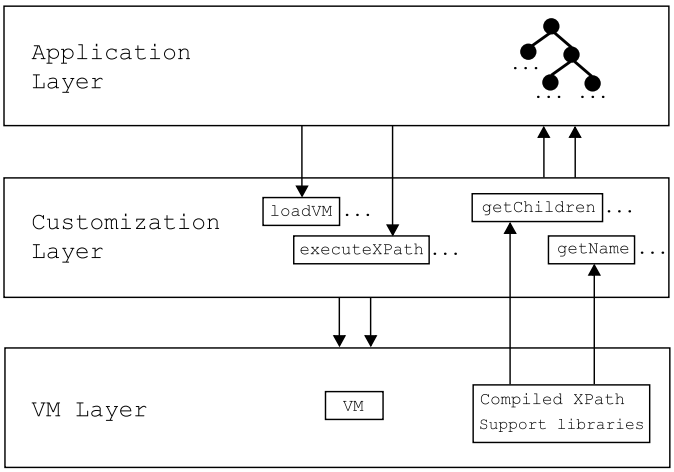
Paper for XML Prague 2007: `http://xmlhack.ru/protva/prague07-generative-xpath.pdf`

Extended abstract proposal for GTTSE'2007: `http://xmlhack.ru/protva/recfree-xpath-abstract-gttse2007.pdf`

Technical implementation

Two parts:

- Compiler to a virtual machine (VM) code
- VM and its runtime environment



The application layer uses the customization layer:

- Load and initialize the VM
- Start XPath evaluation and access the result

The VM layer uses the customization layer:

- Get a collection of nodes (an XPath axis)
- Compare nodes in the document order
- Query node properties, such as name or string value

Example of VM code

```
(define (fac n)
  (if (< n 2)
    1
    (* n (fac (- n 1)))))

(fac 1) ; Evaluates to 1
(fac 6) ; Evaluates to 720
```

It's Scheme R5RS <`http://schemers.org/`>.