XML::Simple

E' pratica abbastanza comune del CPAN quella di creare moduli che ripropongono le funzionalità di altri moduli in versione "semplificata". Basti pensare all'esempio di libwww-perl, che insieme ai vari moduli LWP, fornisce un LWP::Simple che con la sola istruzione **get** permette di scaricare il contenuto di una pagina web.

E' questo il caso di XML::Simple, un modulo ideato per rendere il più immediata possibile la gestione di file XML. Tutte le funzioni del modulo sono praticamente racchiuse nelle due istruzioni **XMLin** e **XMLout**. La prima permette di caricare un file XML in una struttura dati direttamente fruibile da Perl, la seconda di scrivere un file XML a partire da una struttura dati Perl.

Il modulo non implementa di per sè le funzionalità di parsing del formato XML, ma si appoggia a XML::SAX o XML::Parser (il primo dei due che trova installato).

A differenza dei due moduli menzionati, però, XML::Simple esegue tutto in automatico, senza bisogno di scrivere del codice ad-hoc. Il risultato è una struttura dati Perl complessa (generalmente, un hash che contiene vari altri hash e/o array) che riflette il contenuto dei nostri dati XML. Facciamo un piccolo esempio:

```
# il file di esempio (esempio.xml)

<esempio>
  <numero valore="42">quarantadue</numero>
  <stringhe>
      <stringa numero="1">foo</stringa>
      <stringa numero="2">bar</stringa>
      <stringa numero="3">baz</stringa>
  </stringhe>
</esempio>

# lo script che legge il file con XML::Simple

use Data::Dumper;
use XML::Simple;
my $XML = XMLin('esempio.xml');
$Data::Dumper::Indent = 1;
print Dumper($XML);

# l'output dello script

$VAR1 = {
  'numero' => {
    'content' => 'quarantadue',
    'valore' => '42'
  },
  'stringhe' => {
    'stringa' => [
      {
        'numero' => '1',
        'content' => 'foo'
      },
      {
        'numero' => '2',
        'content' => 'bar'
      },
      {
        'numero' => '3',
        'content' => 'baz'
      }
    ]
  }
```

```
};
```

E' possibile andare a manipolare direttamente la struttura ricevuta da **XMLin**, per poi passarla ad **XMLout**, che genererà un nuovo XML con le modifiche da noi effettuate.
Ossia, aggiungendo il seguente codice al nostro esempio precedente:

```
push(
  @{ $XML->{stringhe}->{stringa} },
  { numero => 4, content => "qux" }
);

open(ESEMPIO, ">esempio.xml");
print ESEMPIO XMLout($XML);
close(ESEMPIO);
```

Il risultato sarà un nuovo file esempio.xml che conterrà anche la nostra nuova stringa:

```
<opt>
  <numero valore="42">quarantadue</numero>
  <stringhe>
    <stringa numero="1">foo</stringa>
    <stringa numero="2">bar</stringa>
    <stringa numero="3">baz</stringa>
    <stringa numero="4">qux</stringa>
  </stringhe>
</opt>
```

Ci sono numerose opzioni (molto ben documentate) per affinare il comportamento di XML::Simple. Il modulo non è ovviamente consigliato per lavorare con quantità di dati ragguardevoli, nel qual caso è sicuramente più efficiente costruirsi un parser con XML::Parser o XML::SAX. Il modulo è studiato specificatamente per piccole basi di dati, come ad esempio file di configurazione.