

Information Discovery and Sharing

Nuno Carvalho José João Almeida

Departamento de Informática - Universidade do Minho

16 de Janeiro de 2011



Motivations

- retrieve and store information
 - modular
 - plugins
 - easy to maintain
 - easy to add features

Cross Context

- documentation, source code and other files mining
- discover information about packages
- store information about packages
- provide information to applications and users

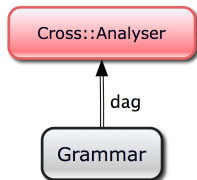
- formalism to represent information
- divided information in facts

$$\begin{aligned} fact &= name \times value \\ value &= atom \times comp^* \\ comp &= (name \rightarrow value)^* \end{aligned}$$

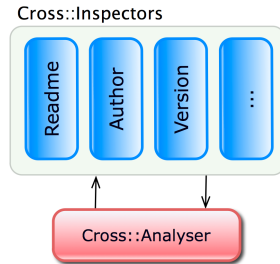
- store in a persistent way
- allow structured processing

$$\begin{aligned} storage &:: subject \times inspector \times fact^* \rightarrow (\dots) \\ retrieve &:: (\dots) \rightarrow fact^* \end{aligned}$$

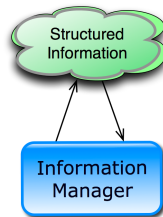
- Directory Attribute Grammar
- write grammars to process files
- run inspectors
- 'typeof' files
- store information



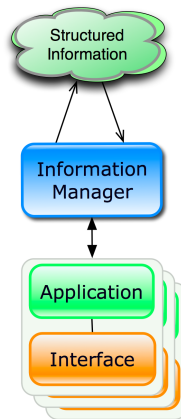
- collection of inspectors
- specialized in retrieving specific information
- modular approach



- store information
- add level of abstraction
- common information interface
- CouchDB backend
- information sharing



- retrieve and process information
- provide information
- user interface
- allow user feedback



The CPAN Ecosystem


```
Package ---> Meta Makefile Readme Changes License Lib*;
```

```
Meta;
```

```
Makefile;
```

```
Readme;
```

```
Changes;
```

```
License;
```

```
Lib;
```



```
Readme {  
  my $i = Cross::Inspector::Readme->new(path=>$_p);  
  my $r = $i->process;  
  
  $db->store('Readme',$r);  
};
```



```
Meta {  
  my $i = Cross::Inspector::Author->new(path=>$_p);  
  my $r = $i->process;  
  
  $db->store('Author', $r);  
};
```

```
Makefile {  
  my $i = Cross::Inspector::Author->new(path=>$_p);  
  my $r = $i->process;  
  
  $db->store('Author', $r);  
};
```



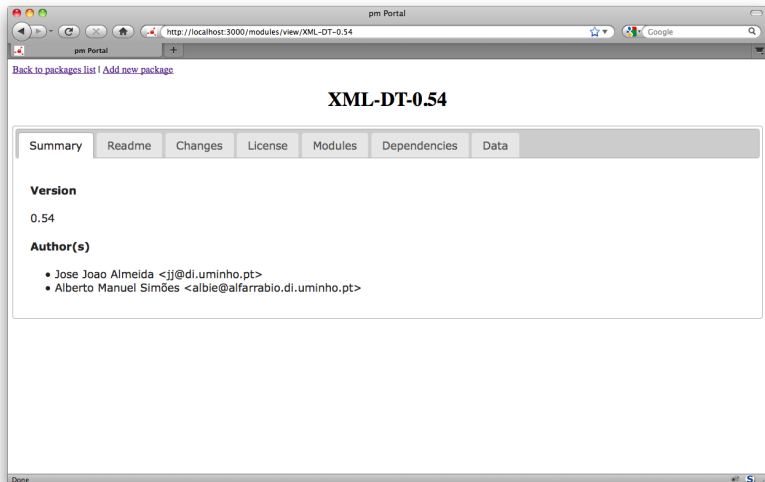
```
Meta {  
  my $i = Cross::Inspector::Version->new(path=>$_p);  
  my $r = $i->process;  
  
  $db->store('Version', $r);  
};  
  
Lib {  
  my $i = Cross::Inspector::Version->new(path=>$_p);  
  my $r = $i->process;  
  
  $db->store('Version', $r);  
};
```



```
Module ---> Meta Makefile Readme Changes License Lib* {  
    $db->store('lib',$_6);  
};
```

```
use Cross::Storage;
```

```
my $db = Cross::Storage->new({package=>$package});  
my $author = $db->retrieve('author');
```



pm Portal

http://localhost:3000/modules/view/XML-DT-0.54

pm Portal

[Back to packages list](#) | [Add new package](#)

XML-DT-0.54

Summary | Readme | Changes | License | Modules | Dependencies | Data

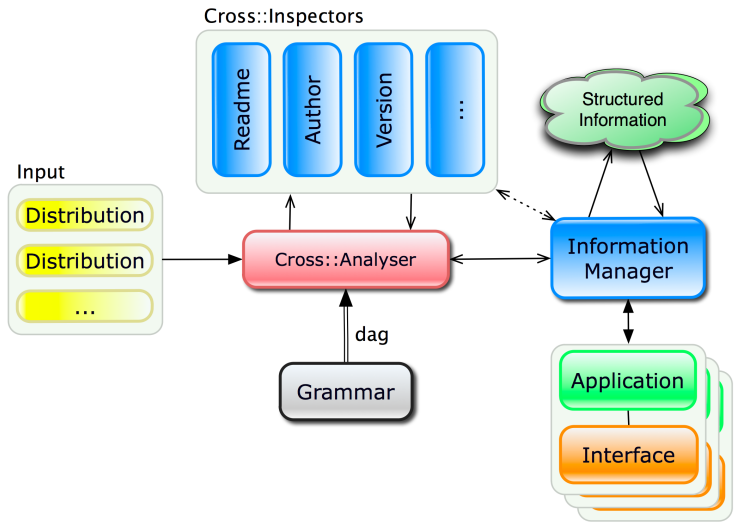
Version

0.54

Author(s)

- Jose Joao Almeida <jj@di.uminho.pt>
- Alberto Manuel Simões <albie@alfarrabio.di.uminho.pt>

Done





- information representation
 - universal type
 - conciliation/union issues
 - information available
 - structured processing
- DAG based analyzers
- Inspectors modular approach
- good results in the CPAN ecosystem
- useful in the CROSS context
 - provide and hare information
 - assist other applications in certification procedures



- tinker information representation formalism
- improve current grammars
- develop new grammars
- improve current inspectors
- implement new inspectors
- improve portal capabilities
- test this approach in other ecosystems
- compile feedback

Thank You!