transformation
languages
all about data processing

- processing formats
- representation
- design, keep logic
theoretical examples
NIMBIN NEWSAGENCY
& MURT
• emailed stories
• uploaded stories
• stories from other news agencies
• etc.
outputs

- newsletter
- website / feeds
- printed articles
- ticker
- etc.
processing

- formats
- ordering
- archiving
- output
other example: models
work flow
models and metamodels

they are everywhere
practical examples
The TXL Programming Language
Source Transformation by Example

- unique programming language
- designed to support source transformation tasks
- quite old
txl: allrounder

- syntax checking / pretty printers
- automation in software maintenance
- supports many kind of transformations
example: tree transformations

- bad (old) html
- want to have strict xhtml
- let txl translate it for us
demo
program transformation

• changing one program into another.

• a big taxonomy
structured object with semantics
stratego/xt

language and toolset for program transformations
Unfortunately no demo
use cases

- java
- php
- java flavours
example: pretty printing java

```bash
$ cat Foo.java
public class Foo {
    public void bar() {
        if(true) {
            System.out.println("Stratego Rules!");
        }
    }
}

$ parse-java -i Foo.java | pp-java
public class Foo {
    public void bar() {
        if(true) {
            System.out.println("Stratego Rules!");
        }
    }
}
```
is this everything?

-> strategic rewriting
stratego / xt
model transformations

model 2 model transformations
metamodeling transformations
Kermeta
example: class2rdbms

class RDBMSModel{
    attribute table : Table[1..*]
}
class Table{
    attribute name : String
    attribute cols : Column[1..*]
    reference pkey : Column[1..*]
    attribute fkeys : FKey[0..*]
}
class FKey{
    reference references : Table
    reference cols : Column[1..*]
}
class Column{
    attribute name : String
    attribute type : String
}

package RDBMSMM;
require kermeta
using kermeta::standard
class Table{
    attribute name : String
    attribute cols : Column[1..*]
    reference pkey : Column[1..*]
    attribute fkeys : FKey[0..*]
}
class FKey{
    reference references : Table
    reference cols : Column[1..*]
}
class Column{
    attribute name : String
    attribute type : String
}
class RDBMSModel{
    attribute table : Table[1..*]
}
conclusions

- helps to deal with different formats/models
- helps to transform different formats/models
conclusions

- (meta-)modeling!
- abstract or concrete transformations
many data formats

different representations

or usages

usage
questions?