BibTEX++
Towards Higher-order BibTeXin

Laura BARRERO SASTRE Fabien DAGNAT Emmanu

Ronan.KERYELL@enst-bretagne.fr

Nicolas TORNERI

—

Laboratoire Informatique & Télécommunic

Département Informatique

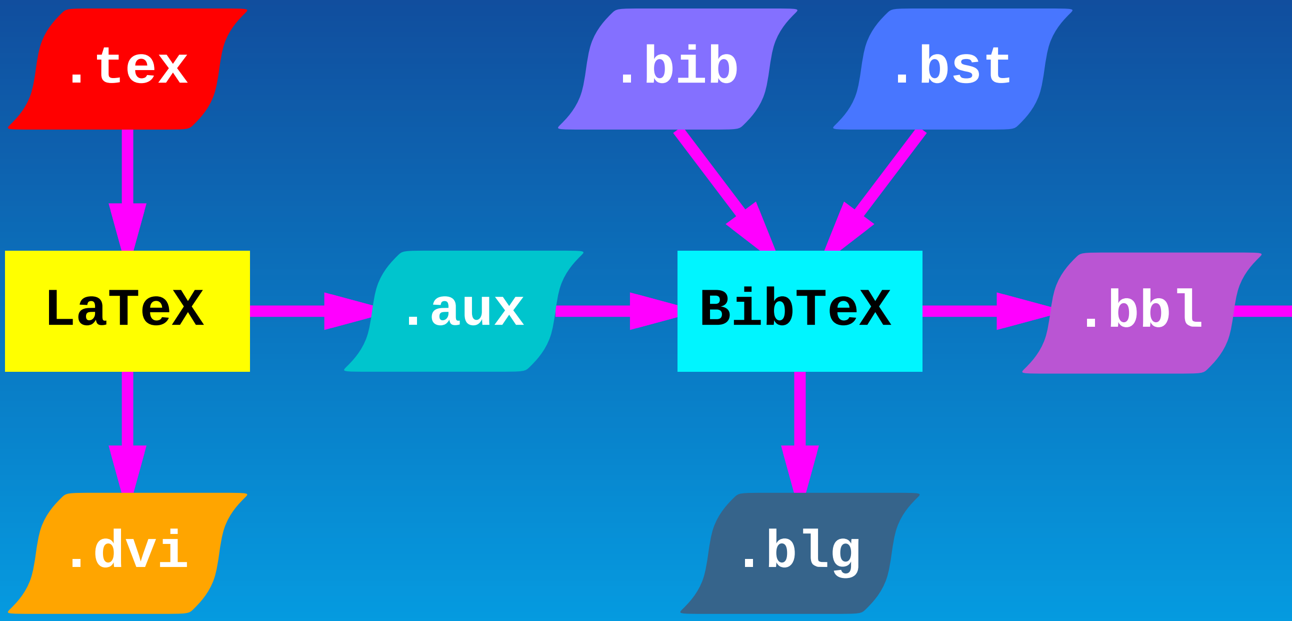
École Nationale Supérieure des Télécommunicatio

26 juin 2003

BibTeX is good for you...

- *THE* bibliographical tool in LaTeX
- Widely used
- Follow logical concepts from the LaTeX world
  - Citation database
  - Bibliography style
  - *Automatic* generation from cited references
  - Typeset further by LaTeX
- Huge existing matter available
  - Large bibliography databases (`http://citeseer.nj.nec.cor`
  - Great amount of styles for many journals, book

BibTEX is good for you...



.tex

.bib

.bst

LaTeX

.aux

BibTeX

.bbl

.dvi

.blg

. . . but. . .

- Old tool : from the 80's (well LATEX too. . . ☺)
- No longer evolves (only improved to accept 8-bit c around 1990)
- Programmable. . . but in an awful 60's stack based (BST) for aliens from the outerspace
  - ▶ Trivial to parse and execute by the computer, e implementation in BibTEX
  - ▶ Just put the burden on the style programmer ☺

```
FUNCTION {sort.format.names}
{ 's :=
  #1 'nameptr :=
  ""
  s num.names$ 'numnames :=
  numnames 'namesleft :=
    { namesleft #0 > }
```

. . . but. . .

```
  { nameptr #1 >
    { " " * }
    ’skip$
   if$
   s nameptr "{vv{ } }{ll{ }}{  ff{ }}{  jj{ }}"
   nameptr numnames = t "others" = and
    { "et al" * }
    { t sortify * }
   if$
   nameptr #1 + ’nameptr :=
   namesleft #1 - ’namesleft :=
  }
 while$
}
```

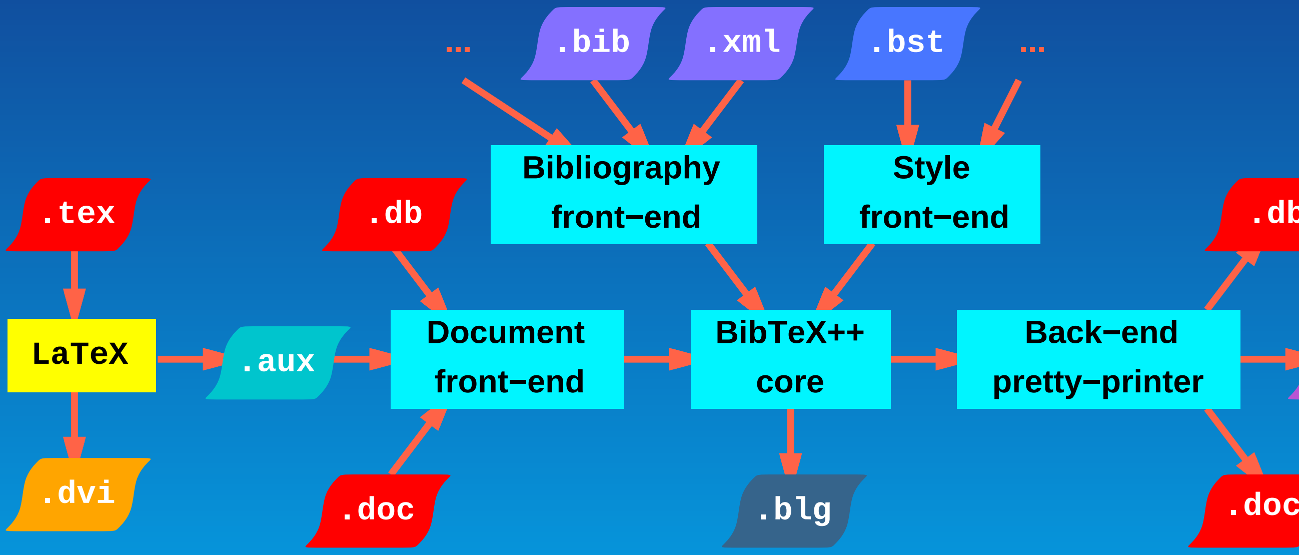
1258 such lines in alpha.bst. . .

...BibTeX++

New needs:

- Multilingual
- UNICODE
- Access to bibliography database from the Interne
- Expressivité du langage de programmation
- Extensibilité non bornée
- Exploitation des styles bibliographiques existants
- Outil générique s'adaptant à d'autres logiciels de
- YAB (Yet Another BibTeX)?

BibTEX++ basic architecture



Adopt the object attitude

- Need to choose a clearer language than BST
- Designing a domain specific language?
  - ▶ Good for selfishness ☺
  - ▶ Yet another (less) cryptic language to learn
  - ▶ With lack of expressiveness?
  - ▶ Oh. . . just improve the language! ☺
- Or use a classical computer language for all the e we want!
- Put all the domain specific stuff in objects: bibliog
- OK since no need for performance
- But into what language?

Programming language

- What we want:
  - ▶ Portable
  - ▶ Clean object support and syntax
  - ▶ Can deal with big programs
  - ▶ Lot of library for all the modern way of life: UNI Internet,. . .
  - ▶ Well known to avoid the *yet another weird-lang* syndrome
- Trade-off
- Let's go for Java

Composant métier

- Fonctions génériques pour générer des bibliograp
- Réécriture en Java des fonctions utilisées dans Bi
- Portable
- Gestion intrinsèque d’UNICODE
- Programmation directe du style bibliographique er expressivité
- Plein de `hook` pour modifier le comportement glob
- Style de message d’information `.blg` conservé

Frontal document

- Transforme les références des citations en format
- Analyse le `.aux` dans le cas de LaTeX
- Parser JLex + JavaCUP
- Peut être un plugin pour rajouter de nouveaux forr

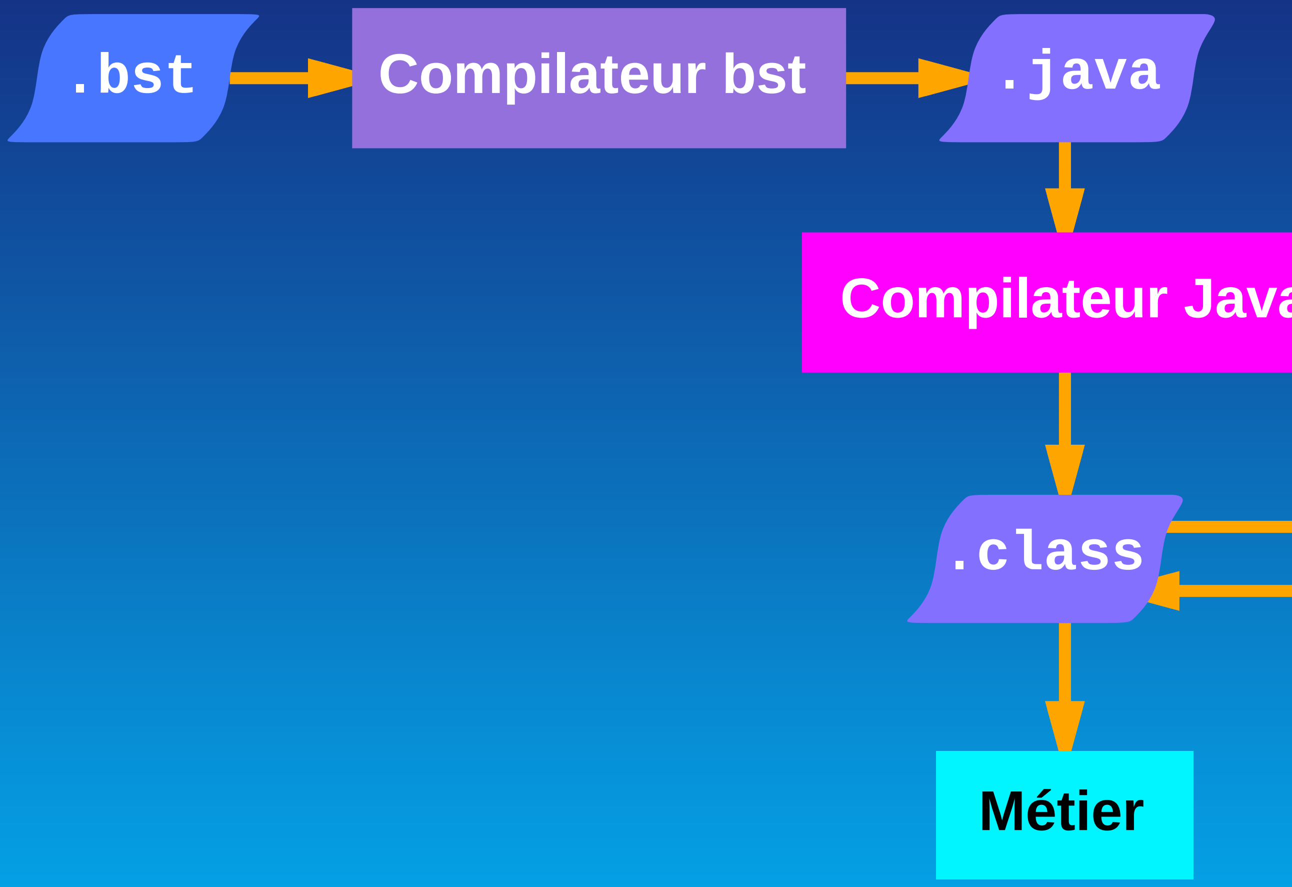
Frontal base de donnée bibliographique

- Récupère base de donnée bibliographique (WWW
- Transforme les références bibliographiques en for
- Analyse le `.bib` dans le cas de LaTeX
- Parser JLex + JavaCUP
- Peut être un plugin pour rajouter de nouveaux forr

Générateur de document bibliographique

- *Prettyprinter* du format interne
- Génère du `.bbl` dans le cas de LaTeX
- Pas difficile à faire
- Peut être un plugin pour rajouter de nouveaux for

Style front-end : a BST compiler



- Big legacy BibTEX style (`.bst`) available

Style front-end : a BST compiler

⇝ Straight traduction of BST code to cle

Rely on advanced compilation and program re-en techniques such as those used in the PIPS projec des Mines de Paris

- Still a SableCC parser
- Use an execution runtime in Java compatible with
- Use a cache of compiled BST styles to speed up c
- ⚠ Well. . . BST is a stack oriented language but J
- Up to this year this stack was still in the generated
- Stack removal is old stuff in computer science but because of efficient JIT JVM implementation

Stack removal

Typing

- BST is not a typed language
- A stack element can hold anything
- But polymorphism is not really used by BST opera a design simplification
- Java is a typed language : nice if lacking BST type infered ⇝ more understandable Java code

Type reconstruction

- Bottom-up approach from BibTEX operators well-k

  `format.name$`

  produces a string on the stack from a string, an in string on the stack
- Propagate all the known types interprocedurally th code
- When ambiguous, keep polymorphic `Cell` objects

Unbalanced stack

- Code transformation from stack based to imperati
  variable
- Assume that each BST block's stack usage ca be
  fixed amount of variables
- What if a BST code usage depend on values? Coc
  with static variable allocation not possible ☹
- Such code does really exist!

  Just in `plain.bst`!

Unbalanced stack

```
FUNCTION {format.names}
{ 's :=  #1 'nameptr :=
 s num.names$ 'numnames :=
 numnames 'namesleft :=
  { namesleft #0 > }
  { s nameptr "{ff~}{vv~}{ll}{, jj}"
      format.name$ 't :=
     nameptr #1 >
      { namesleft #1 >
         { ", " * t * }
         { numnames #2 >
             { "," * }
             'skip$
           if$
           t "others" =
             { " et~al." * }
             { " and " * t * }
```

```
           if$
         }
       if$
      }
       't
    if$
    nameptr #1 + 'n
    namesleft #1 -
  }
 while$
}
```

In the outer `if` the `the`
modify the stack depth
branch push the string
only during the last iter

Unbalanced stack

- Correction:
  - Theoretical answer: just understand the progra
  - Approximation: abstract interpretation + loop p evaluation
  - Real life right now: pattern matching
- Some other usages of stack nasty things: simulati exception-line mechanism with markers on the sta function can throw away an exception that is caug by emptying the stack up to the marker

Plugins

- Permet d'étendre arbitrairement le code
- Dynamique
- Rajout de *hook* dans BibTEX++ : modification de t comportements
- Surcharge de classes
- Permet de nouveaux styles ou de nouvelles sourc bibliographies
- `\bibliography{plugin:ENSTBr/computer-scien` récupère la bibliographie du département informa utilisant un protocole quelconque
  - ▶ `\cite{ENSTBr:keryell88}`
- `\bibliography{plugin:citeseer}` récupère dep

Plugins

`http://citeseer.nj.nec.com/keryell93activity.html`

- `\cite{citeseer:keryell93activity}`
- Donne les entrées BibTEX mais non canonique
- Problème du HTML pas très propre

- `\bibliography{plugin:DBLP}` récupère la bibliog

  `http://dblp.uni-trier.de/`

  - Base en XML `ftp://ftp.informatik.uni-trier.de/pub/users/Ley/bib/re`
  - DTD `http://SunSITE.Informatik.RWTH-Aachen.DE/dblp/db/about/dblp.dtd`
- `\bibliography{plugin:fermivista}`
- `\bibliographystyle{plugin:ENSTBr/computer-`
  charge le style Java `ENSTBr/computer-science-s`

Méta-plugins

- Plugins chargeant d'autres plugins depuis des ser plugins
- Utilise pleinement le chargement dynamique de c
- `\bibliography{plugin:ENSTBr/metaplugin:htt`
- Utilise directement le style depuis le serveur WWW

Security and mobile code (*mobilet*)

Need to extend BibTEX++ with *plugins*, meta-*plugins*, everywhere. . .

¿¿¿What about BibTEX++ virus??? ☹

- Word™: VBScript macros
- HTML browser
  - JavaScript and buggy execution
  - Applet : Java may escape a buggy sand-box
  - All the plugins (*flash*,. . .) in the browser
  - Automatically opened in some mailers (OutLoo
- TEX with a execution *shell* on `\write18` if allowed
- `dvips \special{‘...}` if no `-R`
- PostScript : a true computer language *and* operat

Security and mobile code (*mobilet*)

arbitrary code execution if not in secured mode

- PDF
  - JavaScript
  - Various plugins
  - Can launch a viewer on `http://...` links

⇝ ¡Need to finely control the execution!.

Securing BibTEX++ mobilets

- Written in Java ⇝ freely and heavily relies on Jav
  model
- All actions of a class can be precisely authorized
  `SecurityManager` object : file access, network ac
- Specialization of a `SecurityManager` for a kind of
  styles or plugins
- Mainly only authorized to fill the bibliography cach

Man power on the project

- 1 first year programming project (PAP) in 2000 wit project basis
- 1 first year programming project (PAP) in 2001 wit concept of plugin, preview of stack removal
- 1 third year bibliography study in 2001 on stack re
- 1 master internship on sorting in 2002
- 1 master thesis running on getting all this stuff to months)

What’s next

- Streamline the installation phase
- Plugin mechanism
- Object specialization framework. But what/how?
  - Subclassing
  - Aspect programming
  - Reflection & introspection
  - . . .
- Code transformation framework for automatic loca
  more
- Deal with other worlds than LaTeX: DocBook,. . .
- *Back to typography*: think again to bibliography: h
  complete mix up of Latin, Arabic, Chinese,. . . diffe

What's next

the *same* bibliography?

- Compile BST for other targets : Bibulus,. . .

Conclusion

- It is possible to find computer science research wo world!
- Compatible with BibTEX right now, tested on Linux
- Ready for scalability:
  - Clean portable object oriented language
  - Native UNICODE
  - Recycle legacy dusty deck BST and BIB files th advanced compiler technologies
  - Free software
- Can reach the great unification: for example
  - Word$^{\text{TM}}$ document
  - XML bibliography database from the Internet

Conclusion

- ▶ `.bst` BibTEX for a journal from the Internet

- Tested against all the teTEX distribution. . . Some .
  distribution are wrong! ☹

- First public version this summer

Table of contents

Table des transparents

Introduction