

これからの SATySF_I に望むこと

@Nmatician

2021 年 6 月 26 日

はじめに

自己紹介

- Twitter: @Nmatician
- Github: enunun
- 材料系修士卒（非情報系かつ非プログラマ）
- ソフトウェア開発に関しては素人
 - Github のページにはろくなものはない
- （今のところ）SAT_YSF_I のエンドユーザー
 - 本格的に触りだしておよそ 1 ヶ月
- L_AT_EX と SAT_YSF_I を反復横跳び
 - L_AT_EX とはそれなりに長い付き合い

L^AT_EX と SAT_YSF_I

- SAT_YSF_I は T_EX/L^AT_EX と比較して優位な点も多い
 - 事前の型検査によるエラー報告の精密さ
 - ライセンスがめんどくさくない
- 特に**パッケージ開発のしやすさ**はトツテモスバラシイ
 - 名前空間の分離
 - 「第 0 引数」による周辺の文脈の利用
 - 便利なローカル変数
- 「巨人」たる L^AT_EX を参考にした部分が多い
 - これは変えたほうがいいのか？と思う部分もそれなりに

開発側に望むこと

文書構造の記述方法

- 文書構造は見出しの名前で記述
 - +chapter, +section, +p 等
- 文書構造だけではなく「そのレベルの呼び名」も含む
- 各レベルの「呼び名」は文書構造の記述には不要では？
 - 従属関係のみが本質的なはず
- Markdown では「#」の数で表現
 - じゃあ Markdown 使えば？ □ 表現能力に限界
- あと +p するのがめんどくさい（本音）
- パッケージ製作者にも多大な影響
 - v0.0.x の今のうちに

相互参照における名前空間の分離

- 相互参照はキーと番号の対応を読み取ってなされる
 - .satysfi-aux ファイルにキーと番号の対応が記録
- 「図」や「定理」等の型は記録されず
 - 自動補完させたいときに非常に面倒
 - 同じ数式でも「式」と「力学系」みたく分けたい場合も
- L^AT_EX では cleveref パッケージが有名
 - 読み込み時は \label コマンドにオプション引数が追加
 - キーの名前空間を分離できる
- 要するに cleveref パッケージ相当の機能が欲しい
 - どのレベルで実装？
 - プリミティブか？クラスファイルか？

コミュニティ側に望むこと

開発ノウハウの共有

文書の体裁を変えたい場合、ドキュメントクラスに手を加えるのが「表技」

- L^AT_EX における悲劇 (?) その 1: titlesec パッケージ
- L^AT_EX における悲劇 (?) その 2: authblk パッケージ
 - ドキュメントクラスが担当する機能をパッケージレベルで上書き
 - いわば「裏技」
 - 後者は hyperref パッケージの pdfusetitle オプションと衝突
- ドキュメントクラスに手を加えれば解決だが・・・
 - ライトユーザーにとっては**情報不足**
 - 実装に T_EX 言語や expl3 の知識が要求される可能性
- SAT_YSF_I なら手を加えやすいはず
 - 「手の加え方」がライトユーザーにも周知されるのが望ましい
 - 実践的な例も欲しいよね (後述)

ソースの軽率な公開

- SATySF_I はまだパッケージが少ない
 - 欲しい機能は自分で実装する必要
 - しかしどうやればいいかわからない . . .
 - ソースがなければ解決した人がいてもパクれない
- Github にあるのはパッケージとそのドキュメントが中心
 - もっと実践的な文書作成例が必要
- お前ら PDF だけ挙げるなソースも挙げろ L^AT_EX でも同様だぞ
 - 例：商集合の解説 (<https://github.com/enunun/quoset>)
 - このスライドも (<https://github.com/enunun/satyconf2021>)

ソースの公開先

- Github がおすすめ
 - 個人で文書を書くだけなら add, commit, push だけで十分
- SATySF_I は現在 linguist のサポート外
 - ユニークなリポジトリ数が不足
 - 怪文書を作ってリポジトリを作るだけでコミュニティに貢献！
 - このスライドも貢献にカウント（たぶん）
- 公開するときはライセンスをきちんと設定しよう
 - MIT ライセンスがおすすめ
 - コードをコピペしたときはコピペ元のライセンスに注意

Let's SATySF_I!!