SATYSFI 概説

Takashi SUWA

目次

1. はじめに

SATYSFI は，2017 年度 IPA 未踏事業のひとつとして開発された，新しい組版処理システムである。一口に組版処理システムといっても，大きく分けて

- GUI の画面上に版面がプレヴューされた状態で文書を視覚的に記述していく WYISIWYG エディタ方式
- マークアップ言語によってテキストファイルとして文書を記述し，言語処理系に入力として与えて最終的に PDF などの形式で版面を出力する方式

の 2 つがあるが，SATYSFI は後者にあたる。すなわち，LaTeX や troff などと同様の形式をとるソフトウェアだ。では LaTeX をはじめとする既存のマークアップ言語方式と SATYSFI との違いは何なのだろうか？簡潔に言ってしまえば，最大の違いは

- コマンド定義を記述したコードの可読性が高くカスタマイズしやすい
- ユーザが不適格な入力を与えたときのエラー報告がすばやく，わかりやすく提示される

といった点にある。この性質の詳細を述べる前に，SATYSFI を実装した動機をもう少し詳しく順を追って説明したい。

“マークアップ言語方式と WYSIWYG 方式とのいずれが優れているのか” といった話題は（少なくとも一方に触れる人が多いためか）頻繁に目にし，もはや大喜利の様相を呈しているが，実際にはそれぞれの方式に利点と弱点のトレードオフがあるといってよい。WYSIWYG 方式に比べたマークアップ言語方式一般の利点としては，あくまでも傾向ではあるが

- 入力がテキストファイルなので差分管理が簡単
- ユーザ定義コマンドにより：
  - 複雑な自動処理が実現できる
  - 文書の内容を変更せずに後から体裁を柔軟に変更しやすい

といったことが挙げられるだろう。一方で明白な弱点としては，

- 任意のテキストが入力として渡せるため，ユーザが意図しないコードを入力として与えやすい

というものがある。ユーザが意図していない入力のうち処理系にとっては適格なコードは間違いを検出することが（少なくとも非統計的な手法では）原理的にできないし，これはユーザ自身が出力された版面を見て確かめるしかないが，処理系にとっても不適格なコード，すなわちコマンドの引数の与え方がおかしいといったミスは，処理系が見つけて報告することができる。つまり，ユーザが意図しないコードを与えてしまいやすいという弱点が本質的にあるマークアップ言語という方式では，処理系にとって不適格なコードが入力として渡された場合にどのようなエラー報告を行なうかがユーザの執筆効率に多大な影響をもたらすと考えてよいだろう。

ところが，既存のマークアップ言語方式の組版処理システムでは，この不適格な入力に対するエラー報告がわかりにくいという傾向がある。特に LaTeX の場合を例に挙げると，以下のようなエラーログを数え切れないほど見たユーザは多いだろう：

```
! Undefined control sequence.
```

```
! Missing $ inserted.
```

```
! Missing number, treated as zero.
```

場合によっては原因がすぐわかることもあるが，基本的には「組版処理中に初めて不整合が生じた」ときにエラーが報告される仕組みなので，ユーザの与り知らない，コマンドの実装内の記述と与えた引数の不整合によるエラーが直接報告されることも多い。「第 n 行目ま

でで何かがおかしい」という程度の情報しか得られないことも多く，こうなるともはやコードをコメントアウトしながら二分探索して原因の場所を特定するしかない。

ようやく本題に戻るが，SATYSFI はこれらの問題点を解決するひとつの答えである。具体的にはいわゆる函数型プログラミングに便利な言語をベースに設計されているため

- 強力な型がつく
- グローバルな状態を意識せず局所的な処理にだけ気を遣えばよい

といった函数型の知見の恩恵を享受でき，型検査という形でエラー報告が強化されているのである。

ともあれ，ひとまずエラー報告能力に関しては一旦置いておき，次の第2章では組版処理システムとしてどのような機能が実現されているかについて触れていく。

2. 代表的な機能

2.1. 欧文組版

文字のグリフは正の幅を持ち，紙面には有限の幅がある。当たり前のことであるが，要するにただ文字を水平に並べていくだけでは紙面に収まらないので，どこかで改行しなければならない。「改行」と言うとユーザがテキストデータ上で明示的に改行文字を与えるようにも見えるから，より限定的に「行分割」という言葉も使う。文字の列を組む処理をする際には，この行分割の処理が肝腎要となる。

欧文組版，より一般には “alphabetic な” 文字体系による文字組版では，原則として単語間の空白のどこかで行分割し，各行の長さを両端揃えにする，という組み方が慣習的である。単語間の空白だけでなく，場合によってはハイフネーション，すなわち単語中にハイフンを入れて行分割するという処理も施される。

この欧文組版に関して，SATYSFI は偉大な先駆者である TEX の処理方法に倣い，Knuth–Plass アルゴリズム [Knuth 1978] と呼ばれるアルゴリズムで行分割処理を行なっている。また必要であれば単語中にハイフネーションを施す行分割も自動で行なってくれる。ハイフンを挿入してよい箇所は単語ごとに決まっているが，これに関しては Liang–Knuth アルゴリズムと呼ばれる処理によりハイフネーションパターンの辞書から決定する方法が知られており [Liang 1983]，TEX 処理系に搭載されている。やはり SATYSFI もこれに追従して同様の処理を実現している。

実際に以下の SATYSFI によって組まれた欧文を見てみよう。ところどころ合字処理が施されていたり，自動ハイフネーションが行われているのがわかるだろう：

> The quick brown fox jumps over the lazy dog. ¿But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?
>
> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.2. 和文組版

和文は慣習的に正方形を基調として文字を並べる組版が好まれ，基本的には単語の途中で改行しても構わないなど，欧文組版と比べるとやや緩やかな組版規則に見える。この文書も多くの箇所は日本語で書かれているし，読める方にとっては和文組版は馴染み深いものであろう。ただし，括弧類や句読点類の前後にスペース（アキ）をどの程度の大きさで入れるかといった視覚的調整はそれほど自明ではない。また，いかに改行位置に関して寛容であると言っても例えば「チョコレート」を「チ」と「ョコレート」の間で行分割するとか「チョコレ」と「ート」に分けるといった組み方は強く違和感を喚起してしまうから，行末禁則文字や行頭禁則文字といった取り決めが要請される。

こうした規則をまとめたものとして『日本語組版処理の要件』（W3C 日本語組版タスクフォース，2012）という文書が策定されており，通称として JLreq と呼ばれる。SATYSFI およびその上に実装されたクラスファイル `stdja` および `stdjabook` は，万全ではないもののこの JLreq として定められた和文組版規則に原則従って和文を組むように実装されている。

2.3. 画像挿入機能

画像が挿入できる証拠として，ロゴを図1に掲げておこう。

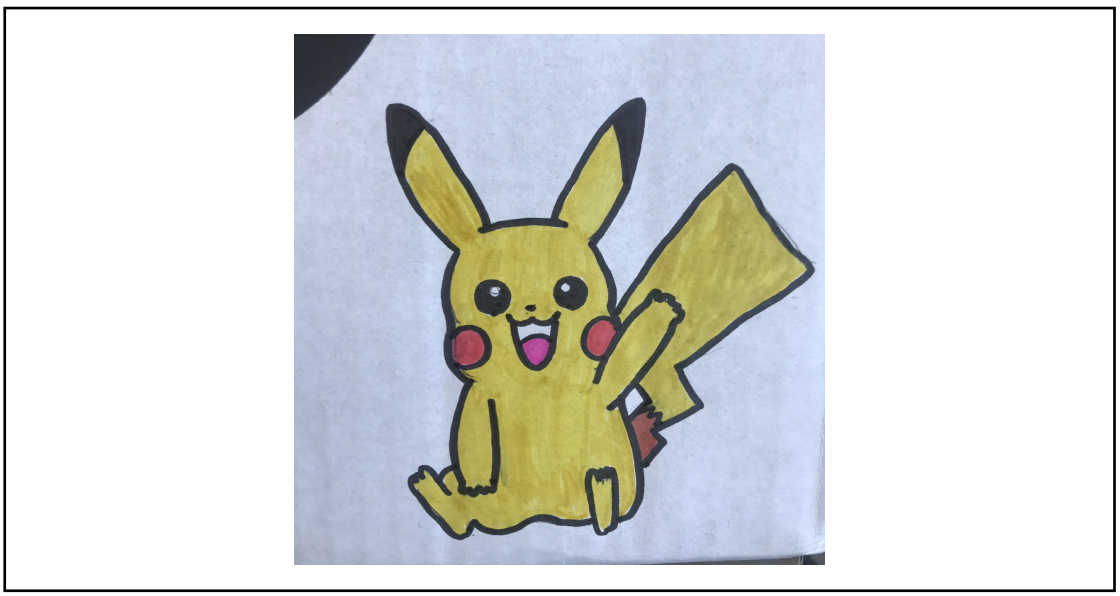

図 1 SATySFi のロゴ

```
+p{
  画像が挿入できる証拠として，ロゴを図 \ref(`fig:logo`); に掲げておこう。
  \figure ?:(`fig:logo`){\SATySFi; のロゴ}<
    +image-frame{\insert-image(7cm)(`satysfi-logo.jpg`);}
  >
}
```

2.4. 表

SATySFi は表組み機能も備えている。コードは長いので省略する。

Program	Answer		Time [s]	
	A	B	A	B
Program 1	Yes	Yes	0.004	0.016
Program 2	No	–	0.004	–
Program 3	Yes	Yes	0.004	0.078
Program 4	Yes	Yes	0.008	0.069

2.5. 数式

既存の組版処理システムである LaTeX が圧倒的な高品質での出力を得意とする機能のひとつに数式組版がある。SATySFi でも，十分にこれに肩を並べられるといってよい程度の数式

組版が実現されている。試しにちょっとした数式を並べてみよう：

$$F = G\frac{Mm}{R^2} \quad \lim_{N\to\infty}\sum_{n=1}^{N}\frac{1}{n^2} = \frac{\pi^2}{6} \quad \Big(\big((A)+B\big)+C\Big) \quad \Big\{\big\{\{A\}+B\big\}+C\Big\}$$

```
+math-list[
  ${F = G \frac{M m}{R^2}};
  ${\limto{N}{\infty} \sums{n = 1}{N} \frac{1}{n^2}
                                          = \frac{\pi^2}{6}};
  ${\paren{\paren{\paren{A} + B} + C}};
  ${\brace{\brace{\brace{A} + B} + C}}
];
```

The solution of the equation $ax^2 + bx + c = 0$ as to x is $x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

```
+p{ The solution of the equation ${a x^2 + b x + c = 0}
    as to ${x} is ${x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}. }
```

$$\frac{\displaystyle\int_0^a x\,\mathrm{d}x \int_0^{\sqrt{a^2-x^2}} r\sqrt{x^2+r^2}\,\mathrm{d}r}{\displaystyle\int_0^a \mathrm{d}x \int_0^{\sqrt{a^2-x^2}} r\sqrt{x^2+r^2}\,\mathrm{d}r} = \frac{2a}{5}$$

```
+math(
  let int1 = ${\int_0^a} in
  let int2 = ${\int_0^\sqrt{a^2 - x^2}} in
  let sqrt2 = ${\sqrt{x^2 + r^2}} in
    ${
      \frac{#int1 x \ordd x #int2 r #sqrt2 \ordd r}{
        #int1 \ordd x #int2 r #sqrt2 \ordd r} = \frac{2a}{5} }
);
```

$$\dfrac{A \qquad \dfrac{B \qquad C}{D}}{E}$$

```
+math(${
  \derive{
    | A | \derive{| B | C |}{D} |
  }{E}
});
```