

論理的文書構造記述のすすめ

守岡 知彦

1 はじめに

ある文書を L^AT_EX を用いて

<pre>1 セクション名 1.1 サブセクション名 本文</pre>

というような形式で出力したいとする。この時、その見掛けに着目して

```
\noindent{\Large\textbf{1 セクション名}}
```

```
\medskip
```

```
\noindent{\large\textbf{1.1 サブセクション名}}
```

```
\smallskip
```

```
本文
```

のように書くことができるが、通常は「1 セクション名」や「1.1 サブセクション名」という部分が節を表していることに着目して

```
\section{セクション名}
```

```
\subsection{サブセクション名}
```

```
本文
```

のように書く。後者の方法を用いることのメリットには

- 番号付けを自動化できる
→ 節の位置やレベルを移動・変更しても手で番号を付け直す必要がない
- 節の書式を自動化できる（文字の位置や大きさ・書体を手で設定する必要がない）
→ スタイルファイルを変更することにより節の書式を一律的に変更できる
- 文書の論理構造が判りやすい
→ 目次を自動生成するなど文書の論理構造に着目した機械処理がしやすい

などがある。このようなメリットは文書の見掛けではなく内容の論理構造に着目して記述していることに起因しているといえ、このような方法をここでは論理的文書構造記述と呼ぶことにする。

2 拡張可能性を利用した論理的文書構造記述

章や節のように L^AT_EX が予め専用の命令を用意しているものの場合、`\chapter` や `\section` といったそうした命令を用いることによって文書内容の論理構造に着目した記述を行うことができる。しかしながら、L^AT_EX が標準ではサポートしていない文書構造の場合、どのように対処すべきであろうか？

前節で述べたように、文書作成・編集作業や文書の機械処理において、文書を見掛けよりも論理構造に着目して記述することにはさまざまなメリットが存在する。そうであれば、L^AT_EX が標準ではサポートしていない文書構造の場合でもやはり論理構造に着目した記述を行いたいものである。

幸い L^AT_EX には利用者が命令や環境などを定義するための機構が存在し、これらを使うことによって L^AT_EX が標準ではサポートしていない文書構造の場合の問題に対処することが可能である。

例えば、注記を

[Note] ここに注記を書く。

のように書きたい場合

```
\newcommand{\note}[1]{
  \begin{quote}
    \textbf{[Note]} #1
  \end{quote}}

```

のような設定を行うことによって、注記を書くための命令 `\note` を定義し、

```
\note{ここに注記を書く。}
```

のように書くことができる。ここで、`\newcommand` というのは新たな命令を定義するための命令で、`{\note}` という部分は定義される命令の名前が `\note` であることを表し、`[1]` という部分は定義される命令の引数の数が1個であることを示す。そして、その後の部分は、定義される命令がどのように展開されるかを示す。ここで、`#1` という部分は引数の値を示す（引数が複数ある場合、第1引数が `#1`、第2引数が `#2` ... 第 n 引数が `#n` となる）。

また、

[Note]

1. 最初の注記
2. 2番目の注記

のように書きたい場合に

```
\newenvironment{Note}{\begin{quote}
  \textbf{[Note]}
  \begin{enumerate}}
{\end{enumerate}}
\end{quote}}

```

のような設定を行うことによって、注記を書くための環境 `Note` を定義し、

```
\begin{Note}
\item 最初の注記
\item 2番目の注記
\end{Note}
```

のように書くことができる。ここで、`\newenvironment` というのは新たな環境を定義するための命令で、`{\Note}` という部分は定義される環境の名前が `\Note` であることを表す（その後には、命令の場合と同様に引数の個数を示すことができるが、今回は引数がないので省略されている）。そして、その後の最初の `{...}` という部分は定義された環境の先頭部分がどのように展開されるかを示し、最後の `{...}` という部分は定義された環境の末尾部分がどのように展開されるかを示す。

3 例題

例えば次のようなテキストの地名、人名を太字にし、かつ、これらと日付に対して索引を作ること考える。

倭國

倭國者、古倭奴國也。去京師一萬四千里，在新羅東南大海中。依山島而居，東西五月行，南北三月行。世與中國通。其國，居無城郭，以木為柵，以草為屋。四面小島五十餘國，皆附屬焉。其王姓阿每氏，置一大率，檢察諸國，皆畏附之。設官有十二等。其訴訟者，匍匐而前。地多女少男。頗有文字，俗敬佛法。並皆跣足，以幅布蔽其前後。貴人戴錦帽，百姓皆椎髻，無冠帶。婦人衣純色裙，長腰襦，束髮於後，佩銀花，長八寸，左右各數枝，以明貴賤等級。衣服之制，頗類新羅。

貞觀五年，遣使獻方物。太宗矜其道遠，敕所司無令歲貢，又遣 [新州刺史高表仁持節往撫之。表仁無綏遠之才，與王子爭禮，不宣朝命而還。至二十二年，又附新羅奉表，以通起居。

日本

日本國者，倭國之別種也。以其國在日邊，故以日本為名。或曰：倭國自惡其名不雅，改為日本。或云：日本舊小國，併倭國之地。其人入朝者，多自矜大，不實對，故中國疑焉。又云：其國界東西南北各數千里，西界、南界咸至大海，東界、北界有大山為限，山外即毛人之國。

長安三年，其大臣朝臣真人來貢方物，朝臣真人者，猶中國戶部尚書，冠進德冠，其頂為花，分而四散，身服紫袍，以帛為腰帶。真人好讀經史，解屬文，容止溫雅。則天宴之於麟德殿，授司膳卿，放還本國。

開元初，又遣使來朝，因請儒士授經。詔四門助教趙玄默就鴻臚寺教之，乃遣玄默闊幅布以為束修之禮，題云「白龜元年調布」。人亦疑其偽。所得錫賚，盡市文籍，泛海而還。其偏使朝臣仲滿，慕中國之風，因留不去，改姓名為朝衡，仕歷左補闕、儀王友。衡留京師五十年，好書籍，放歸鄉，逗留不去。天寶十二年，又遣使貢。上元中，擢衡為左散騎常侍、[鎮南都護。貞元二十年，遣使來朝，留學生 [橘逸勢、學問僧空海。元和元年，日本國使判官高階真人上言：「前件學生，藝業稍成，願歸本國，便請與臣同歸。」從之。開成四年，又遣使朝貢。

ここで、

```
\newcommand{\Name}[1]{\textbf{#1}\index{#1}}
\newcommand{\Place}[1]{\textbf{#1}\index{#1}}
\newcommand{\Year}[1]{#1\index{#1}}
```

という3つの命令を定義し、

`\subsection{倭國}`

`\Place{倭國}` 者，古 `\Place{倭奴國}` 也。去 `\Place{京師}` 一萬四千里，在 `\Place{新羅}` 東南大海中。依山島而居，東西五月行，南北三月行。世與中國通。其國，居無城郭，以木為柵，以草為屋。四面小島五十餘國，皆附屬焉。其王姓 `\Name{阿每氏}`，置一大率，檢察諸國，皆畏附之。設官有十二等。其訴訟者，匍匐而前。地多女少男。頗有文字，俗敬佛法。並皆跣足，以幅布蔽其前後。貴人戴錦帽，百姓皆椎髻，無冠帶。婦人衣純色裙，長腰襦，束髮於後，佩銀花，長八寸，左右各數枝，以明貴賤等級。衣服之制，頗類 `\Place{新羅}`。

`\Year{貞觀五年}`，遣使獻方物。`\Name{太宗}` 矜其道遠，敕所司無令歲貢，又遣 `\Place{新州}` 刺史 `\Name{高表仁}` 持節往撫之。`\Name{表仁}` 無綏遠之才，與王子爭禮，不宣朝命而還。至二十二年，又附 `\Place{新羅}` 奉表，以通起居。

`\subsection{日本}`

`\Place{日本國}` 者，`\Place{倭國}` 之別種也。以其國在日邊，故以 `\Place{日本}` 為名。或曰：`\Place{倭國}` 自惡其名不雅，改為 `\Place{日本}`。或云：`\Place{日本}` 舊小國，併 `\Place{倭國}` 之地。其人入朝者，多自矜大，不以實對，故中國疑焉。又云：其國界東西南北各數千里，西界、南界咸至大海，東界、北界有大山為限，山外即毛人之國。

`\Year{長安三年}`，其大臣 `\Name{朝臣真人}` 來貢方物，`\Name{朝臣真人}` 者，猶中國戶部尚書，冠進德冠，其頂為花，分而四散，身服紫袍，以帛為腰帶。`\Name{真人}` 好讀經史，解屬文，容止溫雅。`\Name{則天}` 宴之於麟德殿，授司膳卿，放還本國。

`\Year{開元初}`，又遣使來朝，因請儒士授經。詔四門助教 `\Name{趙玄默}` 就 `\Place{鴻臚寺}` 教之，乃遣 `\Name{玄默}` 闊幅布以為束修之禮，題云「`\Year{白龜元年}` 調布」。人亦疑其偽，所得錫賚，盡市文籍，泛海而還。其偏使 `\Name{朝臣仲滿}`，慕中國之風，因留不去，改姓名為 `\Name{朝衡}`，仕歷左補闕、儀王友。`\Name{衡}` 留 `\Place{京師}` 五十年，好書籍，放歸鄉，逗留不去。`\Year{天寶十二年}`，又遣使貢。`\Year{上元中}`，擢 `\Name{衡}` 為左散騎常侍、`\Place{鎮南}` 都護。`\Year{貞元二十年}`，遣使來朝，留學生 `\Name{橘逸勢}`、學問僧 `\Name{空海}`。`\Year{元和元年}`，`\Place{日本}` 國使判官 `\Name{高階真人}` 上言：「前件學生，藝業稍成，願歸本國，便請與臣同歸。」從之。`\Year{開成四年}`，又遣使朝貢。

のように書くことができる。

ここで、論理的文書構造記述されたテキスト自体は処理の仕方を指定しておらず、文書内容における抽象的な要素を表現しているに過ぎないことに注意しよう。よって、`\Name` などの命令の定義を変えれば、同じテキストに対して異なる処理を行うことができる。また、このテキストの `\Name{...}` などのパターンをプログラムで処理することもできる。

4 文書構造記述と処理方法の分離

L^AT_EX では命令や環境などを定義することによってさまざまな文書構造を記述することができる。L^AT_EX では形式的に間違っただけの記述に対してはエラーが生じるので、形式的な誤りを検出することもできる。しかしながら、論理的文書構造記述という観点からいえば、幾つかの不満が存在する。

その1つは L^AT_EX の命令や環境が処理方法と不可分であるということである。例えば、テキスト中の人名や地名はさまざまな処理の仕方が考えられるが、それぞれの処理の仕方に応じた命令を定義する必要があり、L^AT_EX で処理するには1種類の命令定義を含まなければならない。即ち、違う処理をするならば、マスターファイルを書き換える必要がある。また、処理と独立に文書構造形式のチェックを行うことができない。つまり、L^AT_EX の命令や環境は文書構造の形式の定義と処

理方法の定義が不可分になっているといえるのである。

XML（とその前身にしてスーパーセットである **SGML**）はこの問題を解決したより一般的・抽象的な論理的文書構造記述を可能とする枠組である。

XML の世界では、文書構造の形式の定義およびそれを用いた検証と、文書構造記述を用いた処理（組版や変換など）は完全に分離されている。文書構造の形式は **DTD** やスキーマ言語というもので定義することができ、それぞれの解析器・検証器で形式的な誤りをチェックすることができる。一方、処理や変換のための枠組としては、XSL などのスタイルシート言語や DOM, SAX などの XML 処理のための枠組などが存在する。

5 おわりに

L^AT_EX を例に論理的文書記述の概念を導入した。**L^AT_EX** では命令や環境などを利用者が定義することにより、標準で用意されていない文書構造も記述することができる。論理的文書記述は見掛けに着目した記述に比べさまざまな利点があり、また、見掛けに着目した処理を妨げるものではないといえる。

このように **L^AT_EX** においても論理的文書記述は可能であるが、文書構造の形式の定義と処理方法の定義が不可分であり、より一般的・抽象的な論理的文書構造記述を行う上では問題があるといえる。これに対し、XML はより一般的・抽象的な論理的文書構造記述を可能とする枠組であり、文書構造の形式の定義と処理方法を完全に分離することができる。

L^AT_EX は人間が直接記述することを前提に設計されており、人間にとっての読みやすさ・書きやすさを考慮した形式であるのに対して、XML は機械処理を優先した形式で **L^AT_EX** に比べやや読みづらく書きづらいといえる。よって、組版・印刷を主要な目的とする場合、**L^AT_EX** が優れており、データ処理や多用途な本格的テキスト・データベースの実現を目指す場合、XML を用いることが優れているといえる。特にデータ処理（データ交換形式）の分野では XML の昨今の普及にはめざましいものがあり、そのためさまざまなツールが登場しつつある。よって、論理的文書記述のための枠組として XML は標準的な地位を築きつつある。

いずれにせよ、なんらかの形で論理的文書記述を行っておけば、多様な処理が可能であるといえる。例えば、**L^AT_EX** と XML を互いに変換することも不可能ではない。一方、論理的文書記述を行う場合、何が記述すべき要素であるかを分析し、それらをどのような形式で記述すべきかということを考える必要がある。特に XML の場合、文書構造の記述と処理が分離されているが為に、実際に処理に用いない文書構造まで記述したくなる誘惑に駆られる（将来使うかも知れないからという理由から）。しかし、詳細に文書構造を記述すればする程労力がかかることを忘れてはならない。