Обзор по извлечению структуры из документов

Богатенкова Анастасия

1 октября 2020 г.

Типы структуры

Существует два типа структуры документов: физическая и логическая.

- Физическая - связана с визуальным представлением документа, то есть как документ разбит на страницы, как страницы разбиты на блоки, блоки на текстовые строки (или изображения) и т. д.
- Логическая - предполагает извлечение структуры, осмысленной для данного типа документов. Так, научные статьи делятся на секции, подсекции и т. д., которые, в свою очередь, могут иметь смысл (введение, список литературы) и делиться на части.

1. Обзор по форматам документов

- Некоторые форматы документов предусматривают хранение информации, связанной с логической структурой документа.
- Некоторые форматы, напротив, хранят информацию только о визуальном представлении документа для пользователя.
- Существуют форматы, хранящие оба типа структуры документов.

SGML и основанные на нем форматы

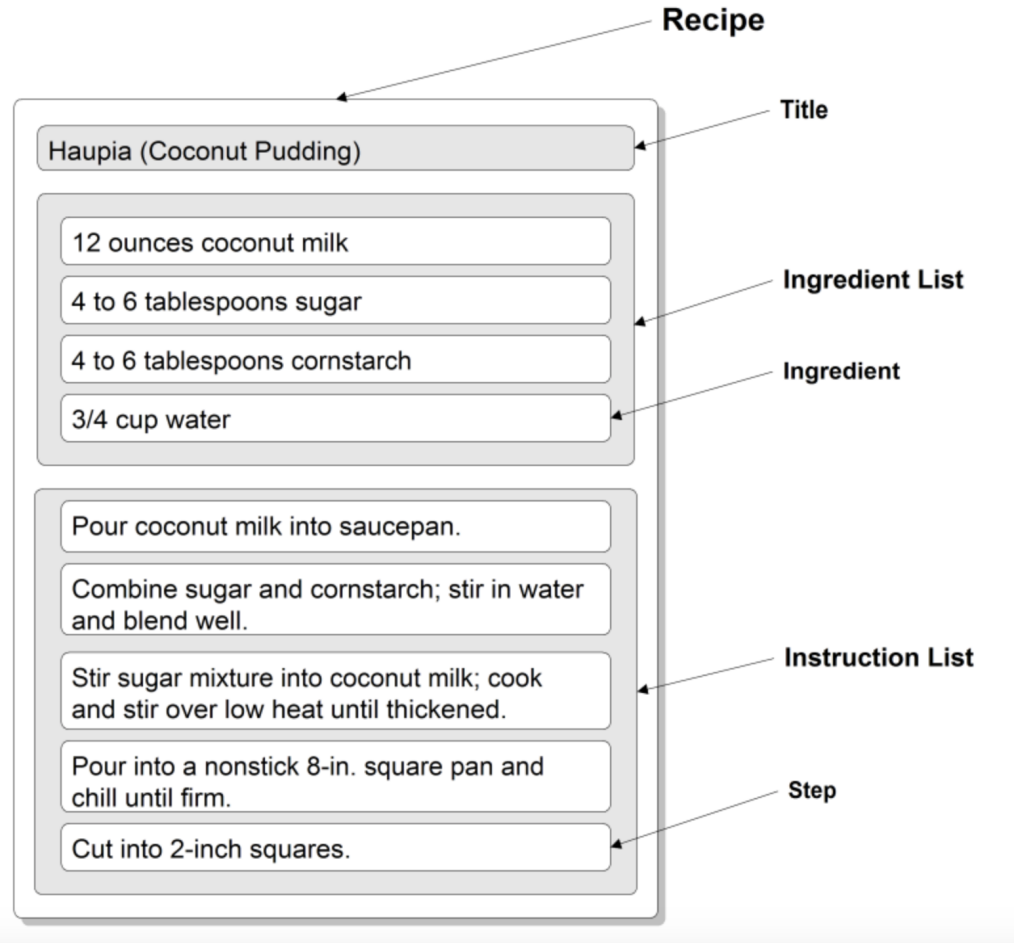
- SGML - язык, использующий разметку (дополнительные аннотации в содержимом документа). В DTD (Document Type Definition) описан тип документа (для каждого типа документа в SGML есть свои правила разметки документов).

+ гибкость и расширяемость

+ много возможностей для программной обработки (но сложно обрабатывать)

– сложность

SGML - пример



Example 1.1. SGML Document for Pudding Recipe

```
<!DOCTYPE recipe SYSTEM "recipe.dtd">
pointer to DTD rules
<recipe
  type="dessert"
  servings="6"
  preptime="10">
<!--I wrote down this recipe
just as my grandmother told me
about it, but I have some doubts.
I need to test it in the kitchen.-->
<title>
Haupia (Coconut Pudding)
</title>
<ingredient-list>
<ingredient>
12 ounces coconut milk
</ingredient>
<ingredient>
4 to 6 tablespoons sugar
</ingredient>
<ingredient>
4 to 6 tablespoons cornstarch<!--Is this
amount correct??-->
</ingredient>
<ingredient>
3/4 cup water
</ingredient>
</ingredient-list>
<instruction-list>
<step>
Pour coconut milk into saucepan.
</step>
<step>
Combine sugar and cornstarch;
stir in water and blend well.
</step>
<step>
Stir sugar mixture into coconut milk;
cook and stir over low heat until thickened.
</step>
&pour-chill-cut;
</instruction-list>
</recipe>
```

DAFS

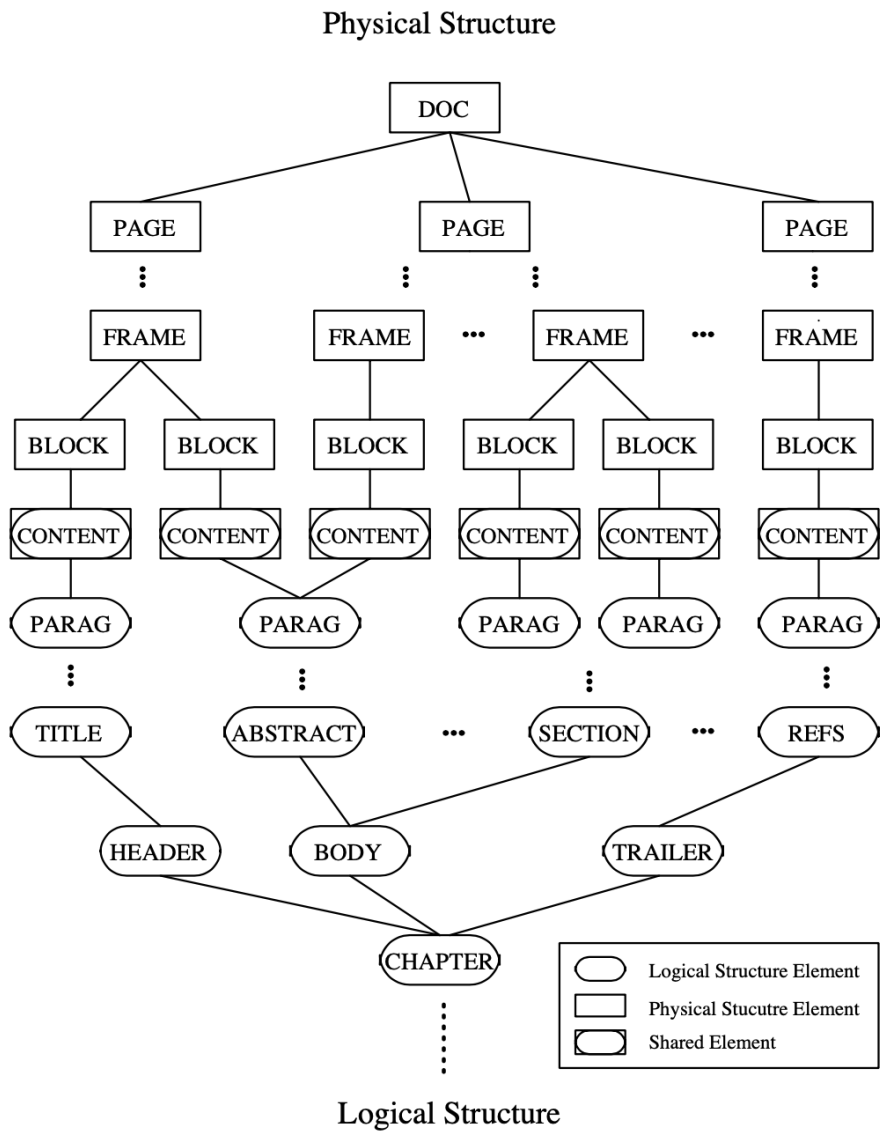
- DAFS предназначен для представления изображений документов и результатов распознавания таких документов. Есть возможность представления нескольких вариантов иерархии для документа.

+ расширяемость

– мало кто умеет обрабатывать (сложность)

DAFS - пример

Документ представляет собой иерархию вложенных друг в друга сущностей, при этом есть два дерева - для логической и физической структуры, листья деревьев с содержимым документа общие.



XML и основанные на нем форматы

+ гибкость и расширяемость
+ интуитивно понятный формат
+ обширная программная поддержка:
    1. стандарты для конкретных доменов (XHTML, SVG, MathML, DocBook)
    2. стандарты, связанные с xml (XSLT, XPath, Namespaces)
    3. программные стандарты (DOM, SAX)
    4. общие инструменты - парсеры на различных ЯП, конвертеры, редакторы, приложения для показа xml.

DocBook

Формат, разработанный в основном для технической документации. Также может быть основан на SGML. Пример из википедии:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
  <title>Very simple book</title>
  <chapter xml:id="chapter_1">
    <title>Chapter 1</title>
    <para>Hello world!</para>
    <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Chapter 2</title>
    <para>Hello again, world!</para>
  </chapter>
</book>
```

Еще несколько примеров, основанных на xml

- DITA (формат, предназначенный в основном для документов, разбитых на темы).
- XHTML 2.0 (расширение html, позволяет добавлять секции и "грамматические"параграфы)
- BNML (Business Narrative Markup Language) (предназначем в основном для текстов законов и договоров)
- 3 xml спецификации для представления документов:
  1. ALTO - для физической структуры (страницы->текстовые блоки->текстовые строки->слова)
  2. TEI - для логической структуры (рекурсивно вложенные друг в друга элементы div, семантический смысл указан в атрибутах)
  3. METS - для отображения между физической и логической структурой.

XML-форматы для графического представления документов

- SVG Для описания двухмерной векторной и смешанной векторно-растровой графики.
  + программно обрабатываеый (как и xml)
  – низкоуровневый, сложно обрабатывать большое количество атрибутов для промежуточного представления.
- XSL-FO Для описания документов, разбитых на страницы.
  + надежная и строгая спецификация
  – слишком большая спецификация (сложно реализовать всё)

ODA

Объектно-ориентированный язык, более сложный, чем SGML, так как помимо логической структуры он описывает также геометрическую структуру и представление документа (как он отображается). Тип документа может быть описан с помощью структуры (наподобие DTD) generic logical structure.

- очень сложный и большой по размеру стандарт
- мало кто умеет обрабатывать

ODA - пример

Содержимое из логической структуры связывается с объектами физической структуры.

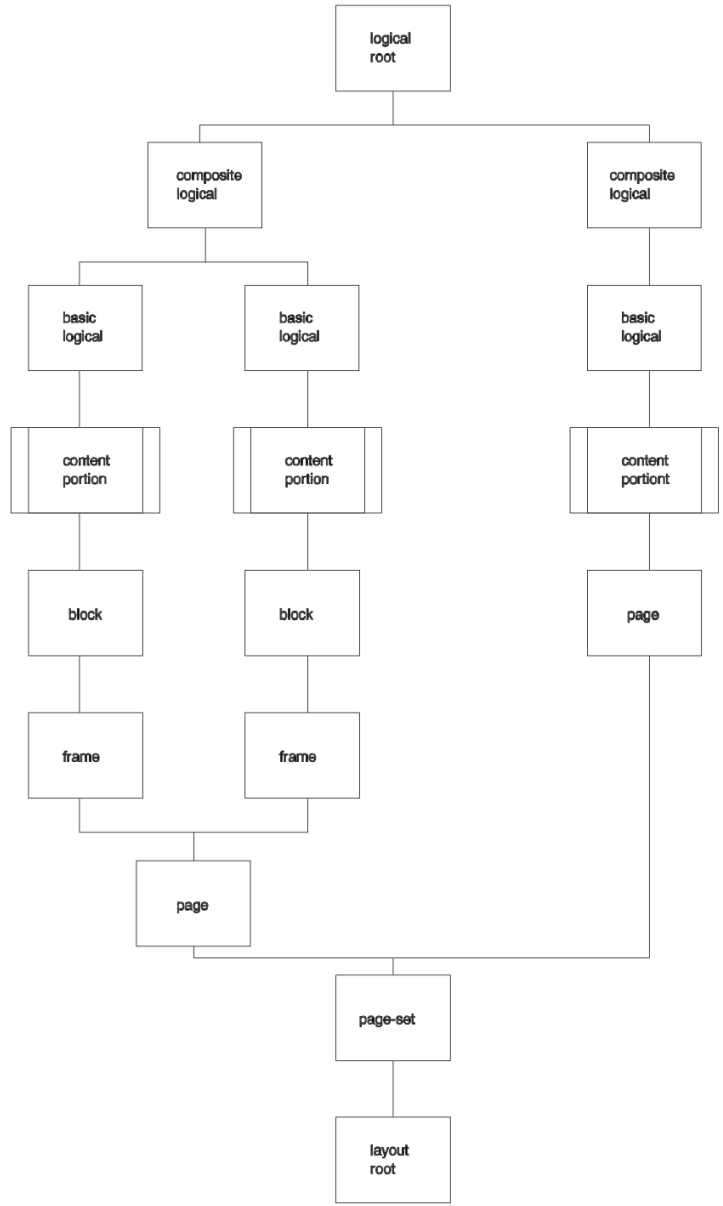


Figure 1: ODA document structures

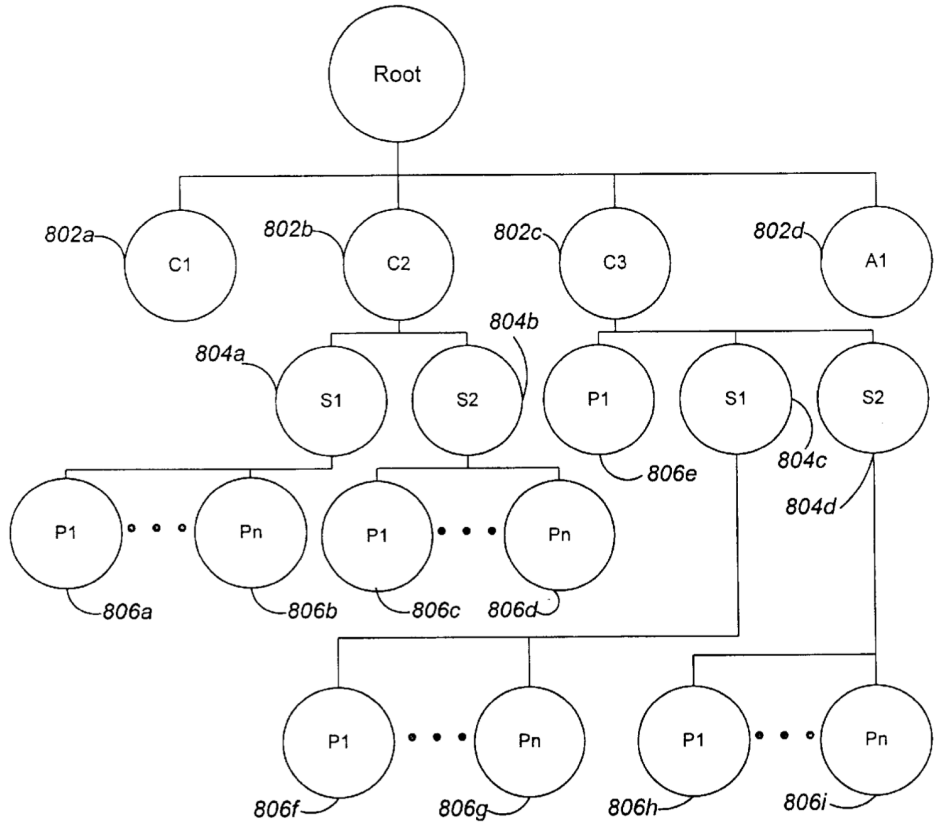
2. Обзор по представлениям структуры документов

Далее будут описаны различные точки зрения на то, в каком виде можно представлять документ:

- в виде дерева;
- в виде графа;
- с использованием формальных грамматик;
- в виде зон и логических меток.

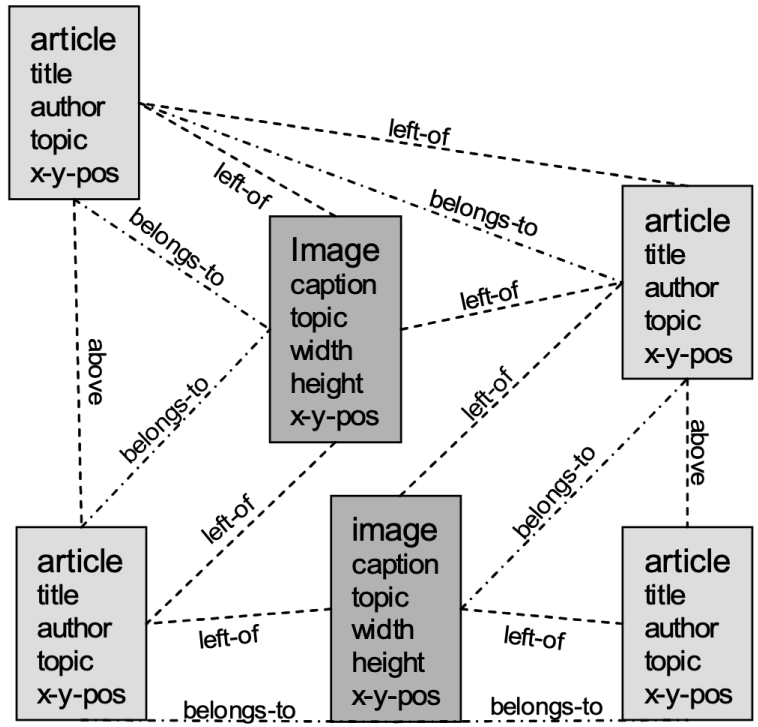
Представление структуры документа в виде дерева

Дерево помогает получить представление документа в виде иерархической структуры, то есть документ разбивается на последовательность вложенных друг в друга элементов.



Представление структуры документа в виде графа

Дерево - частный случай графа. Произвольный граф позволяет представить разбиение документа на части (каждая часть является вершиной графа), а также описать порядок чтения частей (ребра графа могут быть помечены и описывать тип взаимоотношений между частями документа). Структура докумета при этом может получиться не обязательно иерархической.



Представление структуры документа с использованием формальных грамматик

Документ может быть представлен последовательностью правил, которые необходимо обработать с помощью специального парсера. В результате такой обработки получается исходный документ.

[0.5] $< START > \rightarrow < TITLE > < COLUMN > < COLUMN >$
[0.5] $< START > \rightarrow < TITLE > < COLUMN >$
[1.0] $< TITLE > \rightarrow < text\_line >$
[1.0] $< COLUMN > \rightarrow < TEXT\_BLOCKS >$
[0.8] $< TEXT\_BLOCKS > \rightarrow < TEXT\_BLOCK > < space > < TEXT\_BLOCKS >$
[0.2] $< TEXT\_BLOCKS > \rightarrow < TEXT\_BLOCK >$
[1.0] $< TEXT\_BLOCK > \rightarrow < TEXT\_LINES >$
[0.9] $< TEXT\_LINES > \rightarrow < text\_line > < newline > < TEXT\_LINES >$
[0.1] $< TEXT\_LINES > \rightarrow < text\_line >$

(a)

$< START > \rightarrow < TITLE > < COLUMN >$
$\rightarrow < text\_line > < COLUMN >$
$\rightarrow < text\_line > < TEXT\_BLOCKS >$
$\rightarrow < text\_line > < TEXT\_BLOCK > < space > < TEXT\_BLOCKS >$
$\rightarrow < text\_line > < text\_line > < newline > < TEXT\_BLOCK > < space > < TEXT\_BLOCKS >$
$\rightarrow < text\_line > < text\_line > < newline > < text\_line > < space > < TEXT\_BLOCKS >$
$\rightarrow < text\_line > < text\_line > < newline > < text\_line > < space > < TEXT\_BLOCK >$
$\rightarrow < text\_line > < text\_line > < newline > < text\_line > < space > < text\_line >$

(b)

Fig. 6. Representing a document in terms of formal grammars: (a) example of a stochastic context free grammar that derives a text document with a title. Upper case symbols refer to non-terminal symbols, while lower case symbols show terminal symbols. (b) a sample document with a title and three lines, derived using the grammar in (a)

Представление структуры документа в виде зон и логических меток

Документ может быть представлен как плоская структура: последовательность частей какого-либо типа (страницы, текстовые блоки, строки, слова, символы и т. д.) Эти части могут описывать документ в физическом смысле или в логическом. Для представления всей структуры:

- может быть установлена взаимосвязь логических блоков документа с геометрическими;
- документ можно разбить на физические блоки, каждому блоку можно назначить семантическую метку (задача сегментации).

3. Обзор по методам извлечения структуры

Идея: методы извлечения структуры разделить по типу результата, который получается на выходе.