rise to industrial development work. However SGML [1] is a meta-language which permits all desired concepts to be defined, but does not include any pre-defined data type as regards the layout or logical content of a document. Therefore, it was necessary to define by ourselves all types of information required for the complete description of a document content, and their associated characteristics.

So, we had to define an SGML DTD (Document Type Definition) [1] tailored to the description of document recognition results, on completion of the OCR phase.

To the contrary, ODA gives an exhaustive description of the content types in a document, in particular for the layout structure : a page is described as a hierarchical set of frames containing blocks, and each block may contain information of a given nature : characters, vectorial graphics, bitmap images. Precisely it is this type of information that we need. It seemed more advisable to base our work on the types of contents defined in ODA than to start from scratch. As an SGML description of the concepts defined in ODA is available, namely the ODL language [2], we made use of this existing solution [4].

In ODL, a generic layout structure is translated by a series of SGML content models. For documents not comprising any generic structure (that is, for documents whose construction rules are not known), the ODL format provides a minimal DTD which ensures the coding of the main elements defined in ODA.

However, coding the result of document recognition is a reverse task from those who oriented the creation of document layout standards. Therefore, we had considerable adaptation work to do on the ODL DTD to obtain the desired results, that is the coding of all possible layout contents in the recognized documents [5] :

- creating missing types of information in ODA : tables, mathematic formulas, or new attributes for the existing types of information ;
- deleting lots of information useless in our context ;
- changing some information expression modes to simplify and homogenize the syntax.

The resulting DTD is only loosely related with ODL. Therefore we renamed it « ODIL » (Office Document Image description Language).

The second problem is the recognition of the logical structure of a document.

In an SGML context, it consists in grouping recognized elements at layout level and adding structural or semantic information, for shaping them to match the mould of a pre-defined generic DTD corresponding to the processed document type. In this approach, we shall show that the use of the ODIL DTD will facilitate the recognition of the logical structure.

You have to note that ODIL is quite different from previous Document Description Languages, that are generally used to describe generic models for driving document analysis and logical structure recognition [6], [7], [8]. Those approaches are based on top-down or mixed analysis of documents. To the contrary, our PRASAD prototype uses a pure bottom-up approach, and ODIL is used to express the results of recognition without any previous information from a model.

2) Basic elements of ODIL

All layout objects likely to constitute a document are defined in the form of **SGML Elements.** Two types of layout objects are found : basic objects and composed objects. Their characteristics are defined by **SGML Attributes** [1].

A document is described in a tree structure form whose origin is the root of the layout structure of the document so called **ELEMENT dlar**, which stands for « document layout root » [2]. The document begins by a `<dlar>` tag which is mandatory as it contains non optional attributes, and ends by an optional `</dlar>` tag.

It comprises of one or more page groups or pages. The **ELEMENT pages** is optional and refers to a group of pages not used in the current phase of recognition but which may be used later to discriminate a homogeneous set of pages : for example, an article in a magazine...

The **ELEMENT page** identifies a page consisting of information blocks or patches of various types (basic objects) or of frames (composite objects). Besides its representation in ODIL, each page is stored in image form in a multi-image TIFF file : the first image of this file contains the bi-level image of the page less the photos, the second image is a bi-level image containing all halftone photos at their initial location (complement of the previous image), and the successive images are the grey-level photos stored separately.

As explained, five types of information are supported by the ODIL language. The basic objects are zones or blocks, containing homogeneous information (of the same type). Blocks are normally rectangular but in some cases a more complex cut out may be required : the block is then described by a polygon of any shape, in the form of a list of vertices for this polygon. Several separate blocks can be grouped in a frame with a uniform background colour, different from the rest of the page, or outlined by a surrounding border.

In the current state of the project, the text is recognized by OCR process, the photographs are both halftoned and stored separately in grey levels, and all other types of information are considered as graphics and simply kept with the bi-level image of the page. The next step will permit table recognition. For the expression of its results, we shall use the CALS table DTD and include it in ODIL.

The basic objects or blocks are as follows :

ELEMENT cf : object of formatted character type : text block. Text type information logically presents the

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE