

subsection, paragraph and subparagraph.

The idea behind *text* has already been introduced: its creation enables the involuted many-to-many (N:M) *XRef* relationship to be simplified into an entity of type *text* may cite another entity of type *text*. In the absence of generalization, sixteen different relationships would be required to handle all the possible cross references: an entity of type *section*, *subsection*, *paragraph* or *subparagraph* may cite any other entity of type *section*, *subsection*, *paragraph* or *subparagraph*. The idea behind *node* is that a generic identifier can greatly simplify addressing and aggregation operations by removing the need for end-users to know the specific structures involved. The base object in the E-R diagram is *word.placement* whose identifier contains two attributes *all.unit.id*, the symbolic identifier for the generalization *node*, and *word#* the physical position of the word in the unit addressed by *all.unit.id*. The entity-type *word*, representing in effect a word list, is in a one-to-many relationship with *word.placement*. The nature of *all.unit.id* is described later.

### 3.2 Class Structures

The structure of the statutes of figure 2(a) can be viewed as the complex object shown in figure 3. Two types of hierarchy are embedded within the class structure:

- An aggregation hierarchy represented by solid lines to indicate potential groupings of data. A common aggregation will be of *word.placement* to create dynamically any specialization of the generic object *node*. To Sakkinen [1989], this hierarchy represents incidental inheritance.
- An inheritance hierarchy represented by dotted lines to indicate the automatic inheritance of properties (attributes) by lower level objects from higher ones through 'isA' relationships. Thus *text* is a generic object from which the subobjects *section*, *subsection*, *paragraph* and *subparagraph* inherit properties such as text formatting attributes. Other forms of inheritance are for identifiers: textual objects can inherit their identifiers from *node* as described later. To Sakkinen, this hierarchy represents essential inheritance.

Similar rich structures are encountered in other texts such as Shakespeare's plays where the terminology of overlapping fields is used in the humanities to describe the structures. Fields are neither contiguous to each other nor contained completely within one another: lines, stage directions and speeches overlap each other with no clear structure other than that they