SGML/XML & Text encoding

- **ทำไมต้องมีการทำ** encoding text?
- plain text ไม่สามารถให้ข้อมูลอื่นๆที่เราอาจต้องการใช้ เช่น การจัดรูปแบบการพิมพ์
- ต้องมีการใส่ข้อมูลบางอย่างเพิ่มเติมลงไป เพื่อบอกว่าส่วนไหนเป็นตัวหนา ตัวเอียง
- encoding จากแต่ละโปรแกรมก็จะมีลักษณะที่ต่างกัน เช่น MS Word, Word Perfect, Amipro
- การย้ายข้อมูล ต้องมีโปรแกรมช่วยแปลง format

SGML/XML & Text encoding

- ในทาง linguistics เราก็จำเป็นที่ต้อง markup ข้อมูลทางภาษาศาสตร์บางอย่างเข้าไปใน corpus เพื่อใช้ในการวิเคราะห์ทางภาษาศาสตร์
- เช่น ถ้าต้องการศึกษาในระดับ syntax corpus ก็จะต้องมี basic syntactic information เช่น part of speech อยู่ด้วย

linguistic annotation แบบต่างๆ

- **Part-of-Speech tagging**
- This example is taken from the Spoken English Corpus and used the C7 tagset:

  Perdita&NN1-NP0; ,&PUN; covering&VVG; the&AT0; bottom&NN1; of&PRF; the&AT0; lorries&NN2; with&PRP; straw&NN1; to&TO0; protect&VVI; the&AT0; ponies&NN2; '&POS; feet&NN2; ,&PUN; suddenly&AV0; heard&VVD-VVN; Alejandro&NN1-NP0; shouting&VVG; that&CJT; she&PNP; better&AV0; dig&VVB; out&AVP; a&AT0; pair&NN0; of&PRF; clean&AJ0; breeches&NN2; and&CJC; polish&VVB; her&DPS;

- ตัวอย่าง codes ที่ใช้
  - AJ0: general adjective
  - AT0: article, neutral for number
  - AV0: general adverb
  - AVP: prepositional adverb
  - CJC: co-ordinating conjunction
  - CJS: subordinating conjunction
  - CJT: that conjunction
  - DPS: possessive determiner
  - DT0: singular determiner
  - NN0: common noun, neutral for number
  - NN1: singular common noun
  - NN2: plural common noun
  - NP0: proper noun

POS: genitive marker
PNP: pronoun
PRF: of
PRP: prepostition
PUN: punctuation
TO0: infintive to
VB1: be
VM0: modal auxiliary
VVB: base form of lexical verb
VVD: past tense form of lexical verb
VVG: -ing form of lexical verb
VVI: infinitive form of lexical verb
VVN: past participle form of lexical verb

- Syntactic annotation

  คือการ mark ข้อมูลทางด้าน syntax เข้าไปใน text เช่น corpus ของ text ที่ถูก parse แล้วจนได้ tree structure ของแต่ละประโยค หรือเรียกอีกอย่างว่า treebank

  ตย. ประโยค Claudia sat on a stood เมื่อถูก parse แล้วเก็บใน corpus จะอยู่ในรูปของ

  [S[NP Claudia_NP1 NP] [VP sat_VVD [PP on_II [NP a_AT1 stool_NN1 NP] PP] VP] S]

- Semantic annotation

  The example below (Wilson 1996) is intended to give the reader an idea of the types of categories used in semantic tagging:

  And 00000000 the 00000000 soldiers 23241000 platted 21072000 a 00000000 crown 21110400 of 00000000 thorns 13010000 and 00000000 put 21072000 it 00000000 on 00000000 his 00000000 head 21030000 and 00000000 they 00000000 put 21072000 on 00000000 him 00000000 a 00000000 purple 31241100 robe 21110321

The numeric codes stand for:

00000000 Low content word (and, the, a, of, on, his, they etc)

13010000 Plant life in general

21030000 Body and body parts

21072000 Object-oriented physical activity (e.g. put)

21110321 Men's clothing: outer clothing

21110400 Headgear

23231000 War and conflict: general

31241100 Colour

The semantic categories are represented by 8-digit numbers - the one above is based on that used by Schmidt (1993) and has a hierarchical structure, in that it is made up of three top level categories, which are themselves subdivided, and so on.

- Pragmatic annotation

  ตัวอย่างของ pragmatic annotation **เช่น การบอก** anaphoric reference **ดังที่แสดงข้างล่าง**

  1. On Monday (1 the company 1) reported higher fourth quarter

     earnings, raised <REF=1 its dividend and announced plans for a

     3-for-2 stock.

  2. In (7 the "Love Triangle" case 7), <IMP=7(8 the defendant 8) was

     called to give evidence today.

     (from Fligelstone 1992)

- Phonetic annotation

The example below is taken from the London-Lund corpus:

1 8 14 1470 1 1 A 11 ^what a_bout a cigar\ette# . /

1 8 15 1480 1 1 A 20 *((4 sylls))* /

1 8 14 1490 1 1 B 11 *1 ^w\on't have one th/anks#* - - - /

1 8 14 1500 1 1 A 11 ^aren't you .going to sit d/own# - /

1 8 14 1510 1 1 B 11 ^[/\m]# - /

1 8 14 1520 1 1 A 11 ^have my _coffee in p=eace# - - - /

1 8 14 1530 1 1 B 11 ^quite a nice .room to !s\it in ((actually))# /

The codes used in this example are:

end of tone group ^ onset / rising nuclear tone \ falling nuclear tone /\ rise-fall nuclear tone _ level nuclear tone [] enclose partial words and phonetic symbols . normal stress ! booster: higher pitch than preceding prominent syllable = booster: continuance (( )) unclear * * simultaneous speech - pause of one stress unit

การกำกับข้อมูล

- Markup คือข้อมูลอื่นๆที่ใส่เข้าไปในเอกสาร ที่ไม่ใช่ตัวเนื้อหาของเอกสารนั้นๆ
- เดิม markup ใช้หมายถึงเครื่องหมายหรือสัญลักษณ์ต่างๆที่เขียนเพิ่มเติมในเอกสารเพื่อบอกว่าจะให้จัดพิมพ์เอกสารนั้นๆออกมาอย่างไรเมื่อส่งเรียงพิมพ์ เช่น ให้จัดว่าส่วนไหนจะพิมพ์ตัวหนา ตังเอียง ตัวใหญ่
- markup แบบนี้เราเรียกว่า procedural markup เพราะเป็นการกำหนดว่าจะให้ทำอะไรกับข้อมูล ณ ตำแหน่งนั้น ปัจจุบัน markup แบบนี้ก็ยังใช้ในโปรแกรม word processor
- มีปัญหาในการโอนย้ายข้อมูล เพราะ markup คนละแบบ ต้องแปลง file จาก format หนึ่งไปเป็นอีก format หนึ่ง ไม่สามารถแปลงได้ 100%

การกำกับข้อมูล

- Markup อีกแบบเรียกว่า descriptive markup เป็นการ markup เพื่อบอกโครงสร้างของเอกสารนั้นๆ เช่น chapter, section, table of content, เป็นต้น
- ทำให้เอกสารนั้นเป็นอิสระจากสื่อที่จะปรากฏ เมื่อนำเอกสารนั้นไปพิมพ์เป็นหนังสือ ลงในซีดีรอมหรือ เผยแพร่ในอินเตอร์เน็ตก็สามารถทำได้ทันทีโดยไม่ต้องแปลงเอกสารนั้น (โดยจัดรูปแบบ (format) ตามที่ต้องการสำหรับแต่ละสื่อในภายหลัง)
- เอกสารจึงมีลักษณะที่เป็นกลางมากขึ้นทำให้มีการแลกเปลี่ยนข้อมูลกันระหว่างคอมพิวเตอร์ได้ง่ายขึ้น ถ้าไม่ใช้ descriptive markup แต่ไปใช้ procedural markup การแปลงเอกสารจากรูปแบบหนึ่งไปอีกรูปแบบหนึ่งก็จะเสียเวลาและยุ่งยากมากขึ้น

หน้าที่และบทบาทของ TEI

- TEI ย่อมาจาก Text Encoding Initiative เป็น international research project ที่ตั้งขึ้นมาเพื่อพัฒนาและวางแนวทางสำหรับการกำกับข้อมูลเพื่อให้การแลกเปลี่ยนเอกสารอิเล็คทรอนิกส์เป็นไปได้อย่างอิสระโดยไม่มีข้อจำกัดด้านฮาร์ดแวร์คอมพิวเตอร์หรือซอฟแวร์ที่ใช้
- ได้รับการสนับสนุนจาก Association for Computers and the Humanities (ACH), the Association for Computational Linguistics (ACL) และ the Association for Literary and Linguistics Computing (ALLC)
- โครงการของ TEI เริ่มในปี 1987 คู่มือการกำกับข้อมูลภาษาของ TEI ได้รับการเผยแพร่เมื่อเดือนพฤษภาคม ปี 1994
- TEI เสนอให้ใช้ SGML เป็นมาตรฐานในการกำกับข้อมูล ปัจจุบันเปลี่ยนมาสนับสนุน XML (Extensible Markup Language)

SGML (Standard Generalized Markup Language)

- ถูกกำหนดเป็นมาตรฐาน ISO ในปี 1986
- บุคคลที่มีความสำคัญต่อ SGML คือ Charles Goldfarb
- ในปี 1969 ในขณะที่เขาทำงานที่ IBM เขาได้พัฒนา GML (Generalized Markup Language) ร่วมกับ Edward Mosher และ Raymond Lorie เพื่อใช้ประโยชน์ด้าน text formatting, editing และ information retrieval
- ในปี 1978 ANSI ได้ตั้งคณะกรรมการเพื่อวางมาตรฐานภาษาสำหรับใช้กำกับข้อมูล Goldfarb เป็นหนึ่งในคณะกรรมการ และเขาก็ทำงานให้กับ ISO ด้วย โครงร่างฉบับแรกของ SGML จึงได้รับการเผยแพร่ในปี 1985

SGML (Standard Generalized Markup Language)

- SGML เป็นข้อกำหนดที่ใช้กำหนดภาษา (meta-language) คือ SGML เป็นตัวกำหนดว่าภาษาสำหรับใช้กำกับข้อมูล (markup language) ควรจะมีคุณสมบัติอย่างไร
- SGML ช่วยกำหนดโครงสร้างในระดับต่างๆ (hierarchical structure) ของเอกสารนั้นๆได้ว่ามีองค์ประกอบอะไรบ้าง
- SGML เป็น descriptive markup ดังนั้น เอกสารที่ถูกกำกับตามมาตรฐาน SGML จะเป็นกลาง ไม่ขึ้นกับคอมพิวเตอร์ชนิดใดชนิดหนึ่ง ไม่ขึ้นกับอุปกรณ์ที่ใช้เก็บข้อมูล ไม่ขึ้นกับกฎเกณฑ์ใดโดยเฉพาะ และไม่ขึ้นกับงานใดงานหนึ่งโดยเฉพาะ

SGML

- SGML มองเอกสารเสมือนเป็น object อย่างหนึ่ง เอกสารแต่ละประเภท (type) ก็จะมีคุณสมบัติและโครงสร้างที่ต่างกัน เช่น มี section, chapter, paragraph, sentence
- เอกสารจะประกอบด้วยหน่วย (element) ต่างๆ หน่วยเหล่านี้จะต้องถูกกำกับไว้อย่างชัดเจน ในเอกสารนั้น
- โดยทั่วไปก็จะกำกับด้วยแท็กเปิดและปิดท้ายด้วยแท็กปิด เหมือนกับการใช้เครื่องหมายคำพูด (quotation) เปิดและปิดข้อความ
- markup language ที่พัฒนาขึ้นโดยอาศัยมาตรฐานของ SGML เช่น HTML (Hyper Text Markup Language) CDIF (Corpus Document Interchange Format) เป็นภาษาสำหรับกำกับข้อมูลที่ใช้กับคลังข้อมูล BNC

ตัวอย่าง SGML/XML text

```
<anthology>
   <poem><title>The SICK ROSE</title>
    <stanza>
       <line>O Rose thou art sick.</line>
       <line>The invisible worm,</line>
       <line>That flies in the night</line>
       <line>In the howling storm:</line>
    </stanza>
    <stanza>
       <line>Has found out thy bed</line>
       <line>Of crimson joy:</line>
       <line>And his dark secret love</line>
       <line>Does thy life destroy.</line>
    </stanza>
  </poem>
       <!-- more poems go here   -->
 </anthology>
```

(TEI P3: Chapter 2)

องค์ประกอบของเอกสาร SGML/XML

- ในการสร้างเอกสารที่เป็น SGML (SGML document) มีองค์ประกอบอยู่สามส่วนที่ต้องพิจารณา คือ
- 1. SGML/XML declaration
- 2. Document Type Definition (DTD)
- 3. SGML/XML document instance
- SGML/XML declaration เป็นส่วนที่กำหนดข้อมูลเกี่ยวกับชุดตัวอักษรที่ใช้ (character set) และลักษณะโครงสร้างไวยากรณ์ของภาษาที่ใช้กำกับข้อมูล (SGML concrete syntax) ผู้ใช้ทั่วไปจะไม่ต้องยุ่งเกี่ยวกับส่วนนี้ ผู้ดูแลระบบ (system programmer) จะเป็นผู้ดูแลส่วนของ declaration นี้
- ส่วนที่ผู้ใช้ทั่วไปจะต้องสนใจมีเพียง ส่วนที่เป็น DTD และ document instance

Document instance

- คือตัวเอกสารซึ่งจะถูกกำกับด้วยแท็กต่างๆ
- แท็กจะอยู่ในรูปของ <Tag_name> และ</Tag_name> เพื่อบอกจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลส่วนนั้น
  - เช่น <title> TEI Tutorial no 2: SGML </title>
- ในแท็ก อาจจะกำหนด attribute ของข้อมูลหน่วยนั้นด้วยก็ได้
- รูปแบบของแท็กเปิดจะเป็น <Tag_name Att_set> โดยที่ Att_set เป็น list ของ attribute ของแท็กนั้น
- แต่ละ attribute อยู่ในรูปของ Att_name = “Att_value”
- ชื่อของแท็กและ attribute ของแต่ละแท็กจะถูกนิยามไว้ในส่วนของ DTD

Document Type Definition (DTD)

- หน้าที่ของ DTD ก็คือ นิยามชื่อแท็กต่างๆ คุณสมบัติต่างๆของแต่ละแท็กและโครงสร้างของแท็กทั้งหมดที่ใช้
- SGML มอง document เป็น object อย่างหนึ่งซึ่งมีคุณสมบัติประจำตัว
- ดังนั้น เอกสารประเภทเดียวกันก็ควรจะมีคุณสมบัติและโครงสร้างที่เหมือนกัน หรือสามารถกำหนดให้ใช้ DTD เดียวกันได้
- ส่วนเอกสารที่ต่างประเภทกัน เช่น บันทึกช่วยจำ รายงานการประชุม เรื่องสั้น ก็จะมีโครงสร้างและองค์ประกอบที่แตกต่างกัน หรือมี DTD ที่ต่างกัน
- คณะกรรมการของ TEI เป็นผู้ที่ได้วางแนวทางการออกแบบ DTD ของเอกสารแต่ละประเภทเอาไว้
- ผู้ที่จะกำกับข้อมูลภาษาตามมาตรฐาน TEI จึงควรศึกษาดู DTD ที่ TEI ได้กำหนดไว้เพื่อเป็นแนวทางสำหรับสร้าง DTD สำหรับเอกสารประเภทที่ตนเองต้องการ

ตัวอย่างของ DTD

<!ELEMENT anthology (poem+)>

<!ELEMENT poem (title?, stanza+)>

<!ELEMENT title (#PCDATA) >

<!ELEMENT stanza (line+) >

<!ELEMENT line (#PCDATA) >

(TEI P5: Chapter 2)

เดิม SGML สามารถกำหนด optional สำหรับ tag เปิด/ปิด ได้

เช่น <!ELEMENT title - O (#PCDATA) >

- <!ELEMENT anthology (poem+)>
  1 2

  1. ส่วนที่ตามหลัง ELEMENT จะเป็นชื่อของแท็กที่สามารถใช้ได้
  2. ส่วนเนื้อหาบอกว่าหน่วยนั้นประกอบด้วยหน่วยย่อยอะไรบ้าง ในลำดับแบบไหน หรือเป็นข้อมูลประเภทไหน และจำนวนครั้งที่สามารถปรากฏได้

  เครื่องหมาย + หลัง poem หมายความว่า หน่วย anthology จะต้องมีหน่วยของ poem ปรากฏอย่างน้อยหนึ่งหน่วย
- <!ELEMENT poem (title?, stanza+)>

  หน่วย poem ประกอบด้วยหน่วยย่อยสองหน่วยคือ title และ stanza ตามลำดับ โดยที่ title นั้น อาจไม่ปรากฏก็ได้ (เครื่องหมาย ? = optional)

  ส่วน stanza นั้นจะต้องปรากฏอย่างน้อนหนึ่งครั้ง
- <!ELEMENT title (#PCDATA) >

  ภายในหน่วยนี้เป็นข้อมูลที่เป็น #PCDATA (parsed character data) = ข้อมูลจะเป็นตัวอักษรอะไรก็ได้ที่เครื่องรู้จัก (ซึ่งถูกกำหนดเอาไว้ใน SGML declaration)

DTD : Declaration

- DOCTYPE ใช้กำหนดชื่อของ DTD และ declaration อื่นๆ
- DOCTYPE จะเป็นส่วนแรกสุดใน declaration สามารถเขียนได้สองแบบ ตัวอย่างเช่น
  1. <!doctype memo [List_of_declaration]>
  2. <!doctype memo system “c:\memo.dtd” >

- เป็นตัวอย่างการเขียน DTD ของ memo
- แบบ 1. แสดงรายการของการกำหนดรายละเอียดทั้งหมดไว้ภายใน [ ]
- แบบ 2. แยกการกำหนดรายละเอียดทั้งหมดไว้เน file ต่างหากที่ชื่อว่า memo.dtd

DTD : Declaration

- ELEMENT ใช้กำหนดชื่อของแท็ก การละแท็กและเนื้อหาของแท็กนั้นๆ
  - <!ELEMENT name content_declaration >
- content declaration ใช้กำหนดว่าหน่วยนั้นประกอบด้วยหน่วยย่อยอะไรบ้างและมีโครงสร้างอย่างไรหรือมีข้อมูลประเภทไหน
  - ตย <!ELEMENT Memo ((To & From), Body, Close?) >
- กำหนดว่าภายใน Memo มีหน่วยย่อยได้อีกถึง 4 หน่วยคือ To, From, Body, Close
- To และ From จะต้องปรากฏทั้งสองหน่วยแต่หน่วยไหนจะปรากฏก่อนก็ได้แล้วตามด้วย Body และตามด้วย Close ตามลำดับ
- Close นี้อาจจะไม่ปรากฏก็ได้ (เครื่องหมาย ? ใช้บอกว่าไม่จำเป็นต้องปรากฏ)

DTD : Declaration

- ATTLIST ใช้กำหนดคุณสมบัติของแต่ละแท็ก
  - <!ATTLIST Element_name Att_name Declare_value Default_value >
- Element_name เป็นชื่อของแท็ก Att_name เป็นชื่อของ attribute ของแท็กนั้น ค่าของ attribute จะเป็นไปตามที่กำหนดไว้ใน Declare_value ถ้าหากว่าในข้อมูลไม่ได้ระบุค่าของ attribute ไว้ก็จะใช้ค่าที่กำหนดไว้จาก Default_value

  ตย. <!ATTLIST SIG NAME CDATA #IMPLIED
  COMP CDATA #IMPLIED
  TITLE CDATA #IMPLIED>

  SIG มี attribute 3 อย่าง คือ NAME, COMP, และ TITLE
  แต่ละ attribute มีค่าเป็นอะไรก็ได้ (CDATA หมายถึง character data คือข้อมูลอะไรก็ได้ที่เป็นตัวอักษรรวมถึงเครื่องหมายพิเศษ เช่น < > / ด้วย)
  #IMPLIED หมายถึง ไม่จำเป็นต้องกำหนดค่าให้กับคุณสมบัตินั้นก็ได้

DTD : Declaration

- <!ATTLIST poem
  id ID #IMPLIED
  status (draft | revised | published) #CURRENT
  >

  (TEI P3: Chapter 2)

- poem สามารถมี attribute ได้ 2 อย่าง คือ มี id กับ status ค่าของ id คือตัวเลขหรือชื่อเฉพาะที่กำหนดให้กับ poem หน่วยนั้น #IMPLIED หมายความว่า ค่าของ id นี้ไม่จำเป็นต้องระบุก็ได้ ตรงข้ามกับ #REQUIRED ในกรณีนี้ หมายความว่าบาง poem อาจไม่มี id
- ส่วนค่าของ status มีได้ 3 อย่างคือ draft, revised และ published หากไม่กำหนดค่าให้ ก็ให้ใช้ค่าที่มีอยู่ในปัจจุบัน เช่น ถ้าข้อมูลเป็น <poem status="draft"> ..... </poem> ต่อมาเมื่อพบข้อมูล <poem> ..... </poem> ก็ให้ถือว่าค่าของ status มีค่าเป็น draft เหมือนกับ poem ก่อนหน้านี้

DTD : Declaration

- attribute name อาจจะใช้เหมือนกันได้ใน element ที่ต่างกัน
- เราสามารถกำหนด attribute ใน element เพื่อทำ cross reference ได้ ดังในตัวอย่าง
- สมมติว่า มี document instance ดังนี้

```
<poem id='ROSE'>      <!-- Text of poem with identifier 'ROSE' -->
</poem>
<poem id='P40'>       <!-- Text of poem with identifier 'P40' -->
</poem>
<poem>
   <!-- This poem has no identifier -->
</poem>
```

DTD : Declaration

- **และมี DTD ดังนี้**

```
<!ELEMENT poemRef  EMPTY                >
<!ATTLIST poemRef  target IDREF #REQUIRED >
```

- ในตัวอย่างนี้ แท็กที่ชื่อ poemRef ถูกกำหนดให้ content เป็น empty มี attribute ชื่อ target โดยที่ค่าที่เป็นได้คือ ค่าที่ถูกใช้เป็น id (IDREF) และจำเป็นต้องมีการระบุค่านี้เสมอ (เนื่องจากถูกกำหนดให้เป็น #REQUIRED)

- เราสามารถอ้างถึง poem ที่มี id ว่า Rose ได้โดยใช้แท็ก poemRef ดังนี้

```
Blake's poem on the sick rose <poemRef
target='Rose'/> ...
```

DTD : Declaration

- ENTITY ใช้กำหนด entity ที่จะใช้ในเอกสาร
- entity เปรียบเหมือนเป็นชื่อที่เรียกแทนข้อมูลที่กำหนด
- ปกติจะใช้ entity เพื่อประโยชน์ในการเขียนย่อหรือทำ short hand notation สำหรับอ้างถึงตัวอักษรพิเศษหรือสำหรับอ้างอิงถึงเอกสารอื่นๆ
- entity ถูกกำหนดใน DTD โดยการใช้ declaration "ENTITY"
- เวลาที่อ้างถึง entity ในเอกสาร จะใช้เครื่องหมาย "&" นำหน้า และปิดท้ายด้วย ";"
  เช่น "&ent1;”
- เมื่อโปรแกรมที่วิเคราะห์ SGML/XML document พบว่ามีการใช้ entity โปรแกรมจะแทนที่ entity นั้นด้วยค่าเต็มของ entity นั้น
- การใช้ entity ช่วยประหยัดเวลาการพิมพ์ข้อมูลและทำให้สะดวกต่อการดูแลรักษาข้อมูล และช่วยแก้ปัญหาเรื่องตัวอักษรพิเศษได้

DTD : Declaration

- ตัวอย่างของ entity

```
<!ENTITY  CALS   "Computer-aided Acquisition and Logistics
  Support">
<!ENTITY   CHAP1   SYSTEM   "C:\MYDIR\CHAP1.SGM">
```

- เมื่อกำหนด entity CALS แล้ว แทนที่จะต้องพิมพ์ข้อความว่า "Computer-aided Acquisition and Logistics Support" ใน document ทุกครั้งที่ต้องการ ก็สามารถพิมพ์ “&CALS;” เพื่ออ้างถึง entity CALS แทนได้
  เมื่อมีการตีความเอกสาร entity CALS จะถูกแทนที่ด้วยข้อความที่กำหนดนี้
  หรือเมื่อกำหนด entity CHAP1 แล้ว แทนที่จะต้อง ใส่เนื้อความทั้งหมดจากแฟ้มข้อมูล CHAP1.SGM ลงไป ก็สามารถอ้างถึงเนื้อความในแฟ้มข้อมูลนั้นโดย ใช้ entity CHAP1 ได้ ซึ่งเมื่อมีการตีความ เนื้อความทั้งหมดจากแฟ้ม CHAP1.SGM จะถูกนำไปแทนที่ entity CHAP1

DTD : Declaration

- นอกจากการกำหนด entity เพื่อใช้ใน SGML document แล้ว ยังสามารถกำหนด entity ไว้ใช้ในส่วน DTD ได้ด้วย
- แต่การกำหนด entity เพื่อใช้ใน DTD file ต้องเติม % หลังคำว่า entity และใช้ % แทน &
  - <!ENTITY % TEI.prose 'INCLUDE'>

    <!ENTITY % TEI.extensions.dtd SYSTEM 'mystuff.dtd'>

    (TEI P3: Chapter 2)
- เมื่อมีการอ้างถึง entity TEI.prose ภายในส่วนของ DTD entity นี้จะถูกแทนด้วยคำว่า INCLUDE หรือเมื่อมีการอ้าง entity TEI.extensions.dtd ภายในส่วนของ DTD entity นี้ก็จะถูกแทนที่ด้วยเนื้อความจากแฟ้มข้อมูล mystuff.dtd
- entity แบบนี้เรียกว่า parameter entity

Connectors

- Connectors ใช้บอกความสัมพันธ์ของการเกิดร่วมกันของ element ต่างๆ
- connector ที่กำหนดให้ใช้มีสามตัวดังนี้

,	SEQ	all must occur in the order specified
&	AND	all must occur in any order
|	OR	one and only one must occur

- นอกจากนี้ ยังมีเครื่องหมายที่ใช้บอก occurrence indicator อีกสามตัวคือ

+	PLUS	required and repeatable (1 or more times)
?	OPT	optional (0 or 1 time)
*	REP	optional and repeatable (0 or more times)

Content Types

- คือการกำหนดถึงเนื้อหาของแต่ละ element ว่าเป็นข้อมูลชนิดไหน เช่น
- PCDATA (parsed character data) บอกว่า ให้ดูเฉพาะตัวอักษรปกติที่รู้จัก ส่วนเครื่องหมายเฉพาะ เช่น < > จะหมายถึงแท็กจะไม่ถือว่าเป็นส่วนหนึ่งของข้อมูล
- CDATA (character data) บอกว่าข้อมูลเป็นตัวอักษรปกติ เครื่องหมายเฉพาะจำพวก < > หรือ & ก็จะถือว่าเป็นส่วนหนึ่งของข้อมูลด้วย
- RCDATA (replaceable character data) จะเหมือนกับ CDATA ยกเว้น ในส่วนที่ขึ้นต้นด้วย & จะมองว่าเป็นการบอกถึง entity เช่น <a &subk;>&sub+; จะหมายถึง $<a_k>_+$
- EMPTY บอกว่า element นั้นไม่มีเนื้อความข้อมูลอะไร

ตัวอย่าง SGML/XML document

- ลักษณะของแฟ้มข้อมูลที่กำกับด้วย SGML จะขึ้นต้นด้วย <! DOCTYPE ซึ่งจะกำหนด DTD ที่ใช้ จากนั้นจะเป็นส่วนของ document instance ดังต่อไปนี้

```
<!DOCTYPE mydoc SYSTEM "myDoc.dtd" [
<!ENTITY tla "Three Letter Acronym">
<!ELEMENT my.tag - - (#PCDATA)>
<!-- any other special-purpose declarations or
                re-definitions go in here -->
     ]>
<tei.2>
This is an instance of a modified TEI.2 type document,
which may contain <my.tag>my special tags</my.tag> and
references to my usual entities such as &tla;.
</tei.2>
```

(TEI P3: Chapter 2)

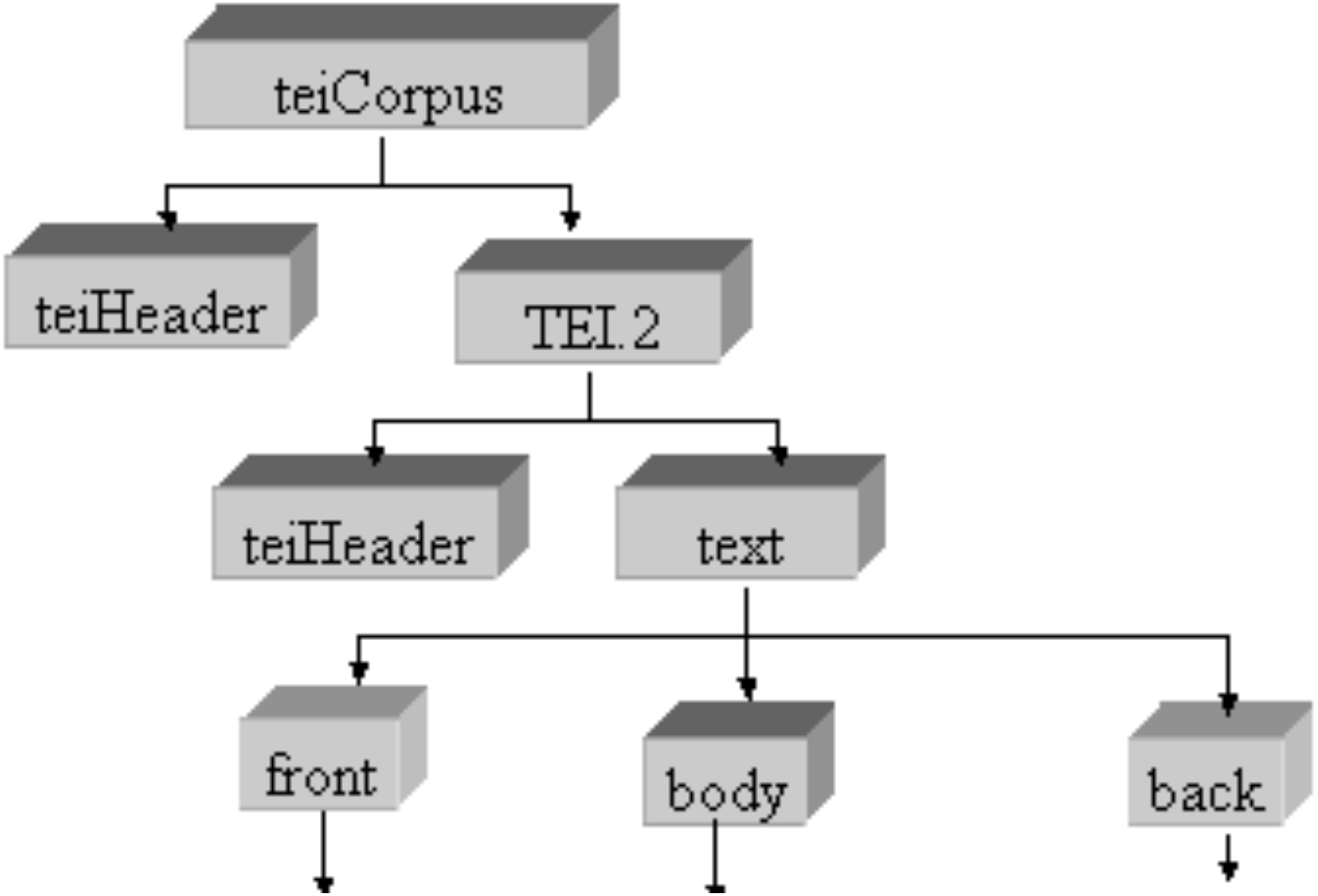
TEI Standard

- The TEI's Guidelines for Electronic Text Encoding and Interchange were first published in April 1994 as two volumes known as TEI P3.
  - SGML was used (defined by ISO 8879)
- TEI P4 appeared in June 2002.
  - Change to XML
  - any document conforming to the original TEI P3 SGML DTD would also conform to the new XML version
- TEI P5, the current version of the TEI Guidelines, was officially released on November 1, 2007

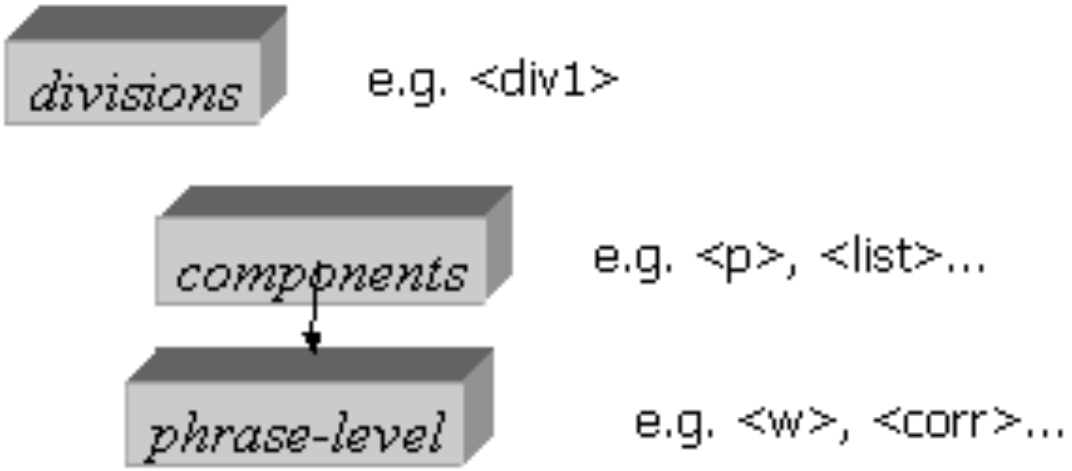
TEI Standard

- Introduction materials
  - Chapter 2 : a gentle introduction to XML
  - TEI Lite. Lite version of the full standard.
- Intended use
  - to provide a standard format for data interchange
  - to suggest principles for the encoding of texts in the same format
  - to support of application-independent local processing.
  - to define a minimal set of conventions for text encoding

Corpus Encoding



- จาก body แยกเป็น component ต่างๆ



- the European projects MULTEXT (LRE) and EAGLES (in particular, the EAGLES Text Representation subgroup), together with the Vassar/CNRS collaboration (supported by the U.S. National Science Foundation), have joined efforts to develop a Corpus Encoding Standard (CES)

- CES เป็น application ของ SGML, XCES เป็น XML version กำลังพัฒนา
- CES สอดคล้องกับข้อกำหนด TEI แต่ extend ให้รับกับงาน corpus encoding ทางด้าน language engineering
- Scope ของ CES
  - Text type : ใช้กับ corpus ที่มี text type หลายประเภท เช่น prose, poem, newspaper, drama, spoken, รวมถึง word list, dictionary
  - Language : ใช้ได้กับ momolingual, multilingual
  - Application : ใช้ encode corpus เพื่องาน language engineering ทุกอย่าง เช่น MT, NLP, lexicography operation ที่ใช้งาน LE เช่น extract sub-corpora; sophisticated search and retrieval รวมถึง collocation extraction, concordance generation; การหา statistics เช่น frequency information, averages, mutual information scores, etc.

- Encoded facts : CES encode **ตั้งแต่ระดับ** discourse **เช่น** paragraphs, chapters **รวมทั้ง** titles, footnotes **และในระดับ** sub-paragraph **เช่น** sentence, quotation, names, terms, abbreviations
  นอกจากนี้ ยัง cover linguistic annotation **ของ** text & speech **รวมถึง** morphosyntactic tagging, parallel text alignment, prosody, phonetic transcription, etc.

- **สรุป** CES provides the following :
  - a set of metalanguage level recommendations
  - tagsets and recommendations for documentation of encoded data;
  - tagsets and recommendations for encoding primary data, including written texts across all genres, for the purposes of corpus-based work in language engineering.
  - tagsets and recommendations for encoding linguistic annotation

Document Structure

- <cesCorpus>
  - <cesHeader type="corpus"> … </cesHeader>
  - <cesDoc>
    - <cesHeader type="text">……</cesHeader>
    - <Text>……</Text> </cesDoc>
    - ….
  - <cesDoc>
    - <cesHeader type="text">……</cesHeader>
    - <Text>……</Text>
  - </cesDoc>
- </cesCorpus>

- 1. Encoding documentation
  - roughly ตรงกับ TEI header
  - แต่ละ text <cesDoc> มี header ของตัวเอง <cesHeader>
  - type= "corpus" หรือ "text" บอกว่าเป็น header ของ corpus หรือ text
  - มี creator, version, status, date.created, date.updated
- <cesHeader> มี 4 element <fileDesc>, <encodingDesc>, <profileDesc>, <revisionDesc>
- 1. <fileDesc> มี element ย่อย
  - <titleStmt>, <publicationStmt>, and <sourceDesc> are required.
  - <titleStmt> ให้ข้อมูล title of the corpus or the individual text and its constituent texts
    <titlStmt><h.title>Jack London's "White Fang": electronic edition</h.title></titleStmt>

– <editionStmt> **ให้ข้อมูลเกี่ยวกับ** particular version of a text.
<editionStmt> Public Domain TEI edition prepared at the
Oxford Text Archive </editionStmt>

– <extent> **ข้อมูลขนาดของ** text
<extent>Filesize uncompressed: 413 Kbytes.</extent>

– <publicationStmt> **ข้อมูล** publication or distribution

<publicationStmt>
<discributor>Oxford Text Archive</distributor>
<pubAddress> Oxford University Computing Services,
13 Banbury Road, Oxford OX2 6NN;
archive@ox.ac.uk
</pubAddress>
<idno>1822</idno>

<availability status=P> <p>This text may not be used to set type for a published edition of the works without the explicit prior permission of the Library of America.

<p>Freely available for non-commercial

use provided that this header is included in its

entirety with any copy distributed</availability>

<pubdate>23 Feb 1993</pubdate>

</publicationStmt>

- <sourceDesc> : **ให้ข้อมูลเกี่ยวกับ** source **ที่นำมา**

  มี tag **ย่อย** <biblFull> **หรือ** <biblStruct>

- <biblStruct> contains a structured bibliographic citation, in which only bibliographic sub-elements appear and in a specified order.
- <biblFull> cotains a bibliographic citation for a text which has been previously encoded in electronic form.

Example :

<**fileDesc**>

< **titleStmt** >

<title>H. G. Wells's "War of the Worlds": electronic edition</title></ **titleStmt>**

<**editionStmt**> Public Domain TEI edition prepared at the Oxford Text Archive </**editionStmt>**

<**extent**> Filesize uncompressed: 357 Kbytes. </**extent**>

<**publicationStmt**> <resp> <role>Distributors</role>

<name>Oxford Text Archive, Oxford University Computing Services, 13 Banbury Road, Oxford OX2 6NN; archive@ox.ac.uk </name> </resp> <idno>1900</idno>

<avail status=P><p>Freely available for non-commercial use provided that this header is included in its entirety with any copy distributed</ avail>

<date>1 Feb 1993</date> </**publicationStmt**>

< **sourceDesc** >

…..Information of the source text <biblStruct>…</biblStruct>

</**sourceDesc**>

</**fileDesc**>

```
<cesHeader version="2.0">
    <fileDesc>
        <titleStmt>
            <h.title></h.title>
        </titleStmt>
        <publicationStmt>
            <distributor></distributor>
            <pubAddress></pubAddress>
            <availability></availability>
            <pubDate></pubDate>
        </publicationStmt>
        <sourceDesc>
            <biblStruct>
                <monogr>
                    <h.title></h.title>
                    <h.author></h.author>
                    <imprint>
                        <pubPlace></pubPlace>
                        <publisher></publisher>
                        <pubDate></pubDate>
                    </imprint>
                </monogr>
            </biblStruct>
        </sourceDesc>
    </fileDesc>
</cesHeader>
```

- 2. <encodingDesc> มี 6 element
  - 2.1 <projectDesc> บอกจุดมุ่งหมายของการ encode

<projectDesc>

The MULTEXT project is assembling a corpus consisting of mono-lingual texts in seven Eastern and Western European languages, together with parallel translations in each of these languages. The original texts were acquired in various forms and marked up for conformance with the MULTEXT/ EAGLES

Corpus Encoding Standard, to test and validate that scheme. MULTEXT has also developed a suite of annotation tools which have been tested on the texts in the corpus.

</projectDesc>

- <encodingDesc>
  - 2.2 <samplingDecl> **อธิบายเหตุผลวิธีการในการ** sampling text

<samplingDecl>

The texts of the core corpus are mostly extracts from books.

The extracts are between 10,000 and 15,000 words long (30 - 40

pages), and are taken from the beginning of the texts. The front matter, prefaces, forewords, list of contents, etc., are not included in the extracts. In some cases, introductions have been left out as well, e.g. introductions by scholars to works of fiction.

Omission of passages in the text may be marked by an <omit> tag.

</samplingDecl>

- <encodingDesc>
  - 2.3 <editorialDecl> บอกรายละเอียดของ editorial principles and practices ในการ encoding of a text.
    <!ELEMENT editorialDecl - - (correction | quotation | hyphenation | segmentation | transduction | normalization | conformance)+ >
    เช่น <segmentation> บอกหลักที่ใช้ในการ segment text
  - 2.4 <tagsDecl> ให้ข้อมูลการ tagging ที่ใช้ใน document.
    <!ELEMENT tagsDecl - - (tagUsage+) >
    tagUsage มี attribute gi บอกชื่อ tag ที่ใช้ occurs บอกจำนวนที่ใช้
  -

```
<tagsDecl>
   <tagUsage gi=name occurs=256>
   <tagUsage gi=div occurs=7>
   <tagUsage gi=head occurs=7>
   <tagUsage gi=p occurs=705>
   <tagUsage gi=reg occurs=2>
   <tagUsage gi=sic occurs=1>
   <tagUsage gi=body occurs=1>
</tagsDecl>
```

- <encodingDesc>
  - 2.5 <refsDecl> specifies how canonical references are constructed for this text.

    <refsDecl>

    A reference system is built up using the identifiers of the following text units: text, division, paragraph, s-unit.
    Each nested division has an identifier which is built up by successively adding to the identifier of the text. Each paragraph has an identifier which adds yet another layer to the immediately superordinate identifier. S-units are numbered within the nearest division, as shown above. After alignment, each s-unit in the core corpus has a "corresp"
    attribute containing a reference to the corresponding unit(s) in the parallel text.

    </refsDecl>

- <encodingDesc>
  - 2.6 <classDecl> contains a series of <category> elements, defining the classification codes used for texts within the corpus.

    <!ELEMENT classDecl - - (taxonomy+)

    >

    <!ELEMENT taxonomy - - (category+ |
    ((h.bibl | biblStruct),category*)) >

    <!ELEMENT category - - (catDesc, category*)

    >

    <taxonomy> กำหนด typology ที่ใช้ classify text

    <category> describe แต่ละ category ใน taxonomy

```
<taxonomy id="b">
  <bibl>Brown Corpus</bibl>
  <category id="b.a">
    <catDesc>Press Reportage</catDesc>
    <category id="b.a1"><catDesc>Daily</catDesc></category>
    <category id="b.a2"><catDesc>Sunday</catDesc></category>
    <category id="b.a3"><catDesc>National</catDesc></category>
    <category id="b.a4"><catDesc>Provincial</catDesc></category>
    <category id="b.a5"><catDesc>Political</catDesc></category>
    <category id="b.a6"><catDesc>Sports</catDesc></category>
    <!-- ... -->
  </category>
  <category id="b.d"><catDesc>Religion</catDesc>
    <category id="b.d1"><catDesc>Books</catDesc></category>
    <category id="b.d2"><catDesc>Periodicals and tracts</catDesc></category>
  </category>
  <!-- ... -->
</taxonomy>
```

- 3. <profileDesc>
  - <creation> contains information about the origination of a text.
  - <langUsage> groups information describing the languages, sublanguages, registers, dialects etc. represented within a text.
  - <wsdUsage> groups information describing the character set(s) used within a text. **ภายในมี** tag <writingSystem>
  - <textClass> groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.
  - <translations> groups information about existing translations of the text.
  - <annotations> groups information about existing annotation files associated with the text. Type = "SEGMENT" **มี** segmentation **ของ** sentence **หรือ** word type="GRAM" **มีการ** mark morpho-syntac info type="ALIGN" **มีการ** align **ข้อมูล**

- Example
- <langUsage>
  <language id="fr" iso639="fr">French</language> <language id="en" iso639="en">English</language> <language id="la" iso639="la">Latin</language> </langUsage>
- <wsdUsage>
  <writingSystem id="ISO 8859-1">ISO character set for western European languages</writingSystem>
  <writingSystem id="ISO 8859-5">ISO character set for Cyrillic</writingSystem> </wsdUsage>
- 4. <revisionDesc> เป็น 4th element ใน header (optional)
  - บอก detail การเปลี่ยนแปลงใน corpus มี tag <change> <changeDate> <respName> (ชื่อผู้รับผิดชอบ) <h.item> บอกลักษณะการ change

```
<cesHeader version="2.0">
    <fileDesc>
        <titleStmt>
            <h.title>Machine-readable version of 1984, ch. 1</h.title>
            <respStmt>
                <respType>typed in and marked with CES tags </respType>
                <respName>A. Student</respName>
            </respStmt>
        </titleStmt>
        <extent>
            <wordcount>6571 </wordcount>
            <bytecount units="bytes">6571 </bytecount>
        </extent>
        <publicationStmt>
            <distributor>Laboratoire Parole et Langage, CNRS</distributor>
            <pubAddress>29, avenue Robert Schuman
                    Aix-en-Provence, France</pubAddress>
            <telephone>+33 42 95 36 33</telephone>
            <fax>+33 42 59 50 96</fax>
            <eAddress>phonetic@univ-aix.fr</eAddress>
            <availability status=restricted>
                internal use only--cannot be distributed</availability>
            <pubDate>6571</pubDate>
        </publicationStmt>
```

```
      <sourceDesc>
          <biblStruct>
               <monogr>
                    <h.title>Nineteen Eighty-four</h.title>
                    <h.author>George Orwell</h.author>
                    <imprint>
                         <pubPlace>New York</pubPlace>
                         <publisher>New American Library</publisher>
                         <pubDate>1949; reprinted 1961</pubDate>
                    </imprint>
               </monogr>
          </biblStruct>
      </sourceDesc>
</fileDesc>
<encodingdesc>
      <projectdesc>
        This English version of the first chapter of Orwell's 1984 is
        encoded for use in the MULTEXT-EAST project. The English is
        to serve as the base for the parallel corpus, and will be aligned
        to versions of the text in Romanian, Bulgarian, Estonian,
        Slovenian, Czech, and Hungarian.
      </projectdesc>
      <editorialdecl>
          <conformance level=1>CES Level 1</conformance>
          <correction status=medium method=silent></correction>
          <quotation marks=none form=std>Rendition attribute values on Q
                and QUOTE tags are adapted from ISOpub and ISOnum standard
                entity set names
          </quotation>
```

```
            <segmentation>Marked up to the level of paragraph plus
                  marking of particular sub-paragraph elements: NAME, DATE,
                  FOREIGN.
            </segmentation>
        </editorialdecl>
        <tagsdecl>
            <tagusage gi=body occurs=1></tagusage>
            <tagusage gi=date occurs=5></tagusage>
            <tagusage gi=div occurs=2></tagusage>
            <tagusage gi=foreign occurs=4></tagusage>
            <tagusage gi=hi occurs=4></tagusage>
            <tagusage gi=name occurs=149></tagusage>
            <tagusage gi=note occurs=1></tagusage>
            <tagusage gi=num occurs=2></tagusage>
            <tagusage gi=p occurs=41></tagusage>
            <tagusage gi=ptr occurs=1></tagusage>
            <tagusage gi=q occurs=22></tagusage>
            <tagusage gi=quote occurs=3></tagusage>
        </tagsdecl>
    </encodingdesc>
    <profiledesc>
        <langusage>
            <language id="fr" iso639="fr">French</language>
            <language id="en" iso639="en">English</language>
            <language id="la" iso639="la">Latin</language>
            <language id="ns">Newspeak</language>
        </langusage>
    </profiledesc>
</cesHeader>
```

- II. Encoding primary data
  - CES กำหนด encoding level ไว้ 3 level
  - level 1 : เป็น minimum encoding level สำหรับ CES conformance ทำ markup ไปถึงระดับ paragraph
  - level 2 : mark element ที่อยู่ในระดับ paragraph
  - Level 3 : mark รายละเอียดมากสุดถึงระดับ sub-paragraph
- Level 1 เป็นไปตาม cesDoc DTD
  - minimum ที่ต้องการ

```
<cesDoc version="3.9">
    <cesHeader version="2.0"> ... </cesHeader>
    <text>
     <body>
       <div> [optional]
         <p>
         <p>
         <p>
         ...
```

- Level 2 มี requirement
  - ผ่านเกณฑ์ level 1
  - ถ้ามีการ mark element ใน sub-paragraph จะต้อง mark ทุก occurrence
  - อักษรพิเศษ สัญลักษณ์ ให้ใช้ SGML entity
  - quotation ถูกแทนด้วย entity หรือ <q> หรือ <quote>
  - recommend ว่า element ใน paragraph level ทุกตัว (lists, quotes, etc.) ควร identify ให้ถูกต้อง, และ <hi> resolve ให้ชัดเจน เช่น <term> <foreign>
- Level 3
  - ผ่าน requirement ใน level 2
  - All paragraph level elements (lists, quotes, etc.) are correctly identified
  - Where possible, <hi> tags are resolved to more precise tags (foreign, term, etc.)

- การ encode name ตาม TEI
  - That silly man <rs key="DPB1" type="person"><name>David Paul Brown</name></rs> has suffered ...
  - That silly man <name key="DPB1" type="person">David Paul Brown</name> has suffered ...
  - That silly man <persName key="DPB1">David Paul Brown</persName> has suffered ... แบบเสนอใน P5
  - <persName><surname>Roosevelt</surname>,<forename>Franklin</forename><forename>Delano</forename></persName>
  - <orgName>The Justified Ancients of Mu Mu</orgName>
  - Mr Frost will be able to earn an extra fee from <orgName type="acronym">BSkyB</orgName> rather than the <orgName type="acronym">BBC</orgName>

TEI P5

- TEI P5 is the current version of TEI
- Instead of using DTD for schema used in XML, Relax NG Schema language is used.
- anthology_p = element anthology { poem_p+ }
  poem_p = element poem { heading_p?, stanza_p+ }
  stanza_p = element stanza {line_p+}
  heading_p = element heading { text }
  line_p = element line { text }
  start = anthology_p
- <!ELEMENT anthology (poem+)>
  <!ELEMENT poem (title?, stanza+)>
  <!ELEMENT title (#PCDATA) >
  <!ELEMENT stanza (line+) >
  <!ELEMENT line (#PCDATA) >

TEI P5

- TEI schemas can be expressed as Relax NG schemas, W3C Schemas, or DTDs
- there is no 'fixed' monolithic one-size-fits-all TEI schema. Instead, *you* are supposed to create *your* own before *you* can start encoding TEI texts.
- Even more important than a schema is the 'blueprint' for *your* TEI schema. ... In TEI world, such a 'blueprint' is just another TEI document with specific elements, and is called an ODD (One Document Does it all).
- customisation is a built-in prerequisite for using TEI. TEI maintains a specific tool for easing this customisation process. It is called 'Roma', and accessible as a user-friendly web form at http://www.tei-c.org/Roma/
- (from http://tbe.kantl.be/TBE/modules/TBED08v00.htm)

TEI P5

The TEI scheme can only be used by customizing it and customizations are also expressed in the ODD language. For example:

```
<schemaSpec ident="myTEIlite">
 <desc>This is TEI Lite with simplified heads</desc>
 <moduleRef key="tei"/>
 <moduleRef key="core"/>
 <moduleRef key="textstructure"/>
 <moduleRef key="header"/>
 <moduleRef key="linking"/>
 <elementSpec ident="head" mode="change">
  <content>
   <rng:text/>
  </content>
 </elementSpec>
</schemaSpec>
```

produces something like TEI Lite, with a slight change

What does an ODD look like?


```
<elementSpec module="spoken" ident="pause">
 <classes>
  <memberOf key="model.divPart.spoken"/>
  <memberOf key="att.timed"/>
  <memberOf key="att.typed"/>
 </classes>
 <content>
  <rng:empty/>
 </content>
 <attList>
  <attDef ident="who" usage="opt">
   <gloss>A unique identifier</gloss>
   <desc>supplies the identifier of the
     person or group pausing.
     Its value is the identifier of a <gi>person</gi>
     or <gi>persGrp</gi> element in the TEI header.
     </desc>
   <datatype>
    <rng:ref name="data.pointer"/>
   </datatype>
  </attDef>
 </attList>
```

... from which we generate


```
element pause  pause.content, pause.attributes
pause.content = empty
pause.attributes =
 att.global.attributes,
 att.timed.attributes,
 att.typed.attributes,
 att.ascribed.attributes,
 [ a:defaultValue = "pause" ] attribute TEIform  text ?
model.divPart.spoken |= pause
att.timed |= pause
att.typed |= pause
att.ascribed |= pause
```

.. or


```
<!ELEMENT %n.pause; %om.RR; EMPTY>
<!ATTLIST %n.pause;
 %att.global.attributes;
 %att.timed.attributes;
 %att.typed.attributes;
 %att.ascribed.attributes;
 TEIform CDATA 'pause'   >
<!ENTITY % model.divPart.spoken
 "%x.model.divPart.spoken; %n.event; | %n.kinesic;
 | %n.pause; | %n.shift; | %n.u;
 | %n.vocal; | %n.writing;">
```

```
<!DOCTYPE bnc SYSTEM  "/home/BNC/SGML/bnc.dtd" [
<!ENTITY % BNCdocs
          SYSTEM "/home/BNC/SGML/bncDocs.ent">
%BNCdocs;
<!ENTITY % BNCchars
          SYSTEM "/home/BNC/SGML/bncChars.ent">
%BNCchars;
<!ENTITY BNChdr SYSTEM "/home/BNC/Texts/corphdr">
]>
<bnc>
&BNChdr;
&ABC;
&ABD;
</bnc>
```

- entities &ABC; &ABD; can be defined in dtd file like these

```
<!ENTITY ABC SYSTEM "BNC/Texts/A/AB/ABC">
<!ENTITY ABD SYSTEM "BNC/Texts/A/AB/ABD">
```

Thai National Corpus

- โครงการสมเด็จพระเทพรัตนฯ ภาควิชารับผิดชอบ
- Use TEI P4 guideline
- เก็บเฉพาะภาษาเขียน comparable กับ BNC
- ดู “tnc.decl”, “tncDoc.dtd”, “corpdhr”, “corptxt” เทียบกับโครงสร้าง BNC
- การนำข้อมูลเข้าผ่านโปรแกรม Tagger, Header
- การกำหนด domain, genre สำหรับจัดประเภทข้อมูลงานเขียน

References

- Developing Linguistic Corpora : a Guide to Good Practice (http://www.ahds.ac.uk/creating/guides/linguistic-corpora/index.htm)
- Text Encoding Initiative (http://www.tei-c.org/index.xml)
- Corpus Encoding Standard (http://www.xces.org/)
- Expert Advisory Group on Language Engineering Standards (http://www.ilc.cnr.it/EAGLES/home.html)
- British National Corpus (http://www.natcorp.ox.ac.uk/)
- Thai National Corpus (http://ling.arts.chula.ac.th/TNC/)