Future Directions

• Few tools exist that capitalize on architectures (SP is the only one we know of!)

• DTD derivation tools—*visual* tools that can derive architecture $A_1$ from base architecture $A$.
  • Allows departments starting from scratch to easily build on already existing architectures.

• Generic processing tools need to be further refined and implemented for processing architecturally conformant documents (e.g. down-translate -> HTML pgm)

• Continue R&D with live data
Example Application

• Corporate Information Management System:
  • System being built internally to manage our documentation using SGML technology (re-use, valid content, automated publishing, decreased time-to-market, etc.)
  • Implementing SGML using our home-grown document/source code repository

• Documents in this repository will conform to various base architectures. Benefits:
  • Generic and fewer processing tools:
    - for processing a class of documents, e.g. for browsing documents conforming to Technical Doc’n architecture, we will use Netscape and a single down-translate -> HTML pgm written for that class.
    - e.g. for processing elements like Graphics / Cross-references
  • Sets the stage for re-use to occur in more predictable, controllable ways.
Content Definition Team (cont’d.)

- Team initially formed to provide way of allowing for information interchange across corporation:
  - We expect interchange to take place more across LOBs, e.g. Wireless <-> PCN, rather than across Base architectures, e.g. Tech Doc’n <-> Marketing
  - Architectures allow for this to be done more easily, e.g. we have prototyped generic tools to do this
Content Definition Team

- Team formed to implement cross-corporate “source” architecture:
  - “Root” architecture from which all other Nortel architectures will be derived
  - Contains all elements deemed applicable to various types of Nortel documents: technical doc’n, marketing doc’n, product information, etc.
  - “Family” of different DTDs on which we can exert some processing control

- Our architectural hierarchy is as follows:
Architectures

- Emerged out of HyTime
- Allows us to define element and attribute class/architecture hierarchies using *architectural forms*:

```xml
<!ELEMENT ABCReqlist - - (ABCReqitem, ABCReqitem+)>
<!ATTLIST ABCReqlist NortelRequirements NAME #FIXED "Reqlist">
```

```xml
<!ELEMENT Reqlist - - (Reqitem, Reqitem+)>
<!ATTLIST Reqlist Nortel Requirements NAME #FIXED "list">
```

```xml
<!ELEMENT list - - (title?, item, item+)>
```

- See [http://www.techno.com/sgmlarch.htm](http://www.techno.com/sgmlarch.htm) for more info
Background

• Nortel—international telecommunications org’n with 65K+ employees

• DTD Issues:
  • Industry standard DTD solution—clearly insufficient
  • Too large and diverse for “mother-of-all-DTDs” DTD solution
  • Let departments create their own DTDs solution => difficult to interchange information

• Hence architectures
  • Allows some control of DTD environment, yet ...
  • Does not constrain individual department’s needs/creativity
  • Control affords us the ability to more easily interchange information across departments
  • Less tools need to be written for document processing; generic tools created for processing documents conforming to specific architecture
  • Valid SGML technique (—no proprietary mechanisms), e.g. SP supports architectural processing
Contents

• Background
• Architectures
• Content Definition Team
• Example Application
• Future Directions
Using Architectures in Nortel