We convened following the ATG2 Plenary. We began our discussions regarding work approach for the week. After much discussion, we decided that we would review the big picture rather than proceed with the spreadsheet review. We commenced by reaffirming the guiding principles defined in Washington. We then reviewed the Naming and Design Rule categories we identified in Washington and prioritized them as follows:

### Initial NDR Categorizations

<table>
<thead>
<tr>
<th>Initial NDSR categories</th>
<th>Priority Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Modularity, Namespaces, and Versioning</td>
<td>VVI</td>
</tr>
<tr>
<td>2. Code Lists and Enumeration</td>
<td>VI</td>
</tr>
<tr>
<td>3. Naming</td>
<td>LI</td>
</tr>
<tr>
<td>4. Type Declarations [simple types, complex types, data types]</td>
<td>VVI</td>
</tr>
<tr>
<td>5. Structuring (to include containership) – local vs. global, qualified vs. unqualified</td>
<td>VVI</td>
</tr>
<tr>
<td>6. Elements or Attributes</td>
<td>VI</td>
</tr>
<tr>
<td>7. Extension, Restriction, and Reuse</td>
<td>VVI</td>
</tr>
<tr>
<td>8. Annotations</td>
<td>LI</td>
</tr>
<tr>
<td>9. CCTS Support</td>
<td>GP</td>
</tr>
<tr>
<td>10. UMM Support</td>
<td>GP</td>
</tr>
<tr>
<td>11. Strict vs. loose Syntax Validation</td>
<td>LI</td>
</tr>
<tr>
<td>12. Content Validation</td>
<td>LI</td>
</tr>
<tr>
<td>13. High Level (other) stuff</td>
<td>LI</td>
</tr>
<tr>
<td>14. Dependency (such as XPath, X Pointer, XLink, XSLT) on other stuff</td>
<td>VVI</td>
</tr>
<tr>
<td>15. Bindings</td>
<td>LI</td>
</tr>
<tr>
<td>16. Processing Instructions</td>
<td>LI</td>
</tr>
<tr>
<td>17. Syntax other stuff</td>
<td>LI</td>
</tr>
<tr>
<td>18. Common Features</td>
<td>LI</td>
</tr>
</tbody>
</table>

**Legend:**

- **VVI** – VERY VERY IMPORTANT
- **VI** – VERY IMPORTANT
- **LI** – LESS IMPORTANT
- **GP** – GUIDING PRINCIPLE
VVI Priority

1. Structuring (to include containership) –
   a. Local vs. global
      i. Qualified vs. unqualified
   b. Schema Construction
      i. Input Form
      ii. Schema structure assembly and containership
2. Type Declarations [simple types, complex types, data types]
3. Modularity, Namespaces, and Versioning
4. Extension, Restriction, and Reuse
5. Dependency (such as XPath, X Pointer, XLink, XSLT) on other stuff

We came up with the following items as being important in our discussions:

◆ Reusable components for CCT, CC and BIE + message schemas (could call other components)
◆ Context independent components (CCT & CC) more important than context specific (BIE and messages)
◆ Implementation reusability (DB, GUI, etc) --> reusable component within schema
◆ Reuse at meta level - reuse at implementation level are a question of quality of the production rules.
◆ From a user perspective the availability of a schema is most important

We agreed that the CEFACT XML NDR document is a tool to provide the following ATG2 deliverables:

◆ XML representation of reusable context independent components (CCT's & CC's)
◆ XML representation of reusable context specific components (BIE's)
◆ XML schemas for messages and objects using BIE's

We also agreed that the XML representation of a given component should be reusable in other XML schemas. This implies that we are defining a modular approach for XML design.

We also agreed that we need to define:

◆ a method for representing CC's in XML and
◆ a methodology for extending and restricting a "standard schema", i.e. allow for industry specific invoice schemas to be derived form the published UN/CEFACT standard invoice schema.

Issue: Who does the extended version - ATG or an outside organisation?
In order to capture all of the rules applicable for a component we need to capture more rules than what can be expressed in an XML schema, i.e. business rules expressed as props.

Issue: Should these normative rules be captured as annotation in the XML schema?

Issue: We need to resolve the basic issue of are we developing a standard vs. interoperability of components of the schema.

We agree on:

◆ Type reuse shall be supported

We don’t agree yet on:

◆ If Element reuse should be supported
◆ If one of our deliverables should be a library of element names or do we rely on the underlying BIEs
◆ If we should use standard business terms versus CC dictionary entry names

**CC/BIE Representation**

A question was raised on the floor – How are we going to represent CCs and BIEs? There seemed to be some agreement that CCs should be represented by Types and BIEs may be represented by Types or Elements, however there was no formal decision made. We had a discussion on CC binding to XML. Gunter Stuhec presented his CCT/XML paper from UBL. Everyone agreed to review the paper and prepare for a future discussion on this topic.

**Structure Discussion**

We looked at a number of constructs: Russian Doll, Salami Slice, Venetian Blind and Garden of Eden (from UBL). We quickly discounted Russian Doll and Salami Slice as not suited for our needs. We discussed Venetian Blind and Garden of Eden at great length. We also defined a new approach that we called various names including Venetian Garden and Garden of the Blind. This approach is:

```
/- Any level neutral reusable complexType/simpleType declarations
\- Name types

/- Message Level global elements
\- Interoperability/Forward compatibility (abstract type and xsi:type)

- complexType management to abstract type on extension/restriction from "my" namespace versus "other" namespaces.
```

A summary of our pros and cons for the two leading candidates (Venetian Blind and Garden of Eden):
Venetian Blind – Global Types, Local Elements, May be qualified

Pros
a. Binding reflects OO serialization (.Net, JAVA)
b. Supports type reuse
c. Well accepted
d. Makes for short tag names

Cons
a. Does not support element reuse (even if namespace qualified) except indirectly through binding the whole type in which it is declared to an element in the outer schema reuse
b. Allows for the same element to be bound to multiple types
c. Allows the same element name to be used for multiple semantically non-equivalent elements
d. Requires type aware processor to be able to conclude the semantic equivalence of two elements
e. Makes reuse of XPath based logic difficult (XSLT and content based logic)

Garden of Eden – Global Elements, Globally Declared Types, FullyQualified

Pros
a. Instance document is semantically unambiguous
b. An element in the instance points to the type
c. Elements and types are reusable
d. Does not require type aware processors
e. Makes reuse of XPath based logic easy
f. Supports building TBG required library
g. Makes mapping between EDIFACT and XML possible at the element level

Cons
a. Not well known
b. Makes for long tag names
c. Increases file size
d. Does not directly support OO serialization
e. Does not support long term direction of IT
f. The ability to extend using both type and element can create standardization issues
g. Creates many more tag names than venetian blind (some estimates are in the hundreds of thousands)

Much discussion on merits/drawbacks of each. Mark had mini meeting with TBG1 and laid out pros and cons of both. Consensus of TBG1 was:

♦ Global elements are preferred over local
♦ Semantic clarity in the instance document without reliance on tools such as XPath is a requirement
The tag names should be harmonized with the CCTS naming conventions
ATG should be delivering a semantically unambiguous glossary of tags

We took a straw poll and the vote was 6 to 4 in favor of the venetian blind approach with 3 abstentions and 3 supporters of global absent from the room. We decided that it was important to enter into the UBL dialogue on this subject. We will also discuss on our conference calls and look for resolution in London.

**Agreed Upon Naming and Design Rules**
We reached agreement on the following NDR rules:

- **Processing Instructions MUST NOT be used**
- **The Nillability attribute MUST NOT be used**
- **Wildcards MUST NOT be used**
- **Two schemas shall be developed for each standard. One schema shall be a run-time schema devoid of documentation. Once schema shall be a fully annotated schema that employs XHTML for the annotations.**
- **Mixed Content MUST NOT be used**
- **Built-in Simple Types SHOULD be used where possible**
- **Simple Type restriction MAY be used wherever possible**
- **Union technique MAY be used to merge datatypes**
- **Complex Types MUST be named**
- **The absence of a construct or data MUST NOT carry meaning**
- **Substitution groups MUST NOT be used**
- **Attribute Groups MAY be used**
- **ID/IDREF MUST NOT be used**
- **The XSD prefix MUST be used. xmlns:xsd= [http://www.w3.org/2001/XMLSchema]**
- **The XSI prefix SHALL be used where appropriate**
- **Abstract Complex Types MAY be used.**
- **(not finalized) Complex Type extension SHOULD be used where appropriate**
- **(not finalized) The ‘final’ attribute shall be used to control**
- **(not finalized) The ‘block’ attribute shall be used to control**
- **Complex Type restriction SHOULD be used**
- **The ‘final’ attribute SHALL be used to control**
- **The ‘block’ attribute SHALL be used to control**
- **Key/KeyRef May be used to**
- **Notations MUST NOT be used**
- **UpperCamelCase (UCC) MUST be used for naming elements and types**
- **lowerCamelCase(ICC) MUST be used for naming attributes**
- **‘un’ will be used as the namespace token for UN/CEFACT.**

Gunther as our editor will begin to develop an outline and draft of the ATG NDR document containing these and other agreements reached this week
**UML2XML Project**
Thomas gave short update on UML2XML. After much discussion we agreed that the UML2XML project must necessarily wait on the NDR project. However Thomas will open discussions with OAG regarding the progress of UML 2.0 and the expected flexibility of the next version of XMI. Gunter will have a discussion with Mario Jeckle (Daimler Chrysler) and Dave Carlson to see if they would be interested in participating.

**Generic Header Project**
We had a 1 hour presentation from the Generic Header Team. We agreed to work with them on the XML aspects of the generic header project.

**Final Session**
It looks as if UBL will merge with ATG. UBL NDR will open up it's conference calls for ATG members. Suggestion is interim ATG2 meeting simultaneously with UBL in London week of April 28, hosted by APACS. A majority of members were willing to participate in an interim work session in London. We agreed to maintain bi-weekly conference calls, extended by 30 min in order to allow topics to be discussed. Every other call at 10 AM eastern or 6 AM eastern. Next call will be on March 24 at 10 AM eastern then April 7 at 6 AM. We reiterated that anyone can participate in the work of ATG - we should try to increase participation to include X12, UBL, OAG, and others.