**IBIS Common Format**

Version 0.1  
Created: 20090613  
Latest Edit: 20090613  
Author: Jack Park

TODO:  
- Expand on the explanations of a Compendium serialization  
- Articulate the rules for assigning link types  
- Trim unnecessary attributes  
  - Subject to dialogues

**Background**

A DTD that specifies the XML serialization for IBIS elicitation platforms is specified. The specification has been created based on dialogues among the following individuals related to particular IBIS platforms:  
- Jack Park—Compendium  
- David Price—DebateGraph  
- Mark Klein—Deliberatorium

**Differences and Restrictions**

There are known differences among the three platforms. Those differences suggest certain restrictions to capabilities as follows:  
- Common Node Types  
  - Question (Issue)  
  - Answer (Position)  
  - Map  
  - Pro Argument  
  - Con Argument  
- ELEMENT = node ATTRIBUTE = "label"  
  - Restricted to 70 characters

**Behaviors**

**Nested Maps**

Compendium is known to permit *nested maps*. That is, Compendium captures maps as *view elements*, each view serving as a container for the nodes captures in the map node for which it stands. Other platforms are free to treat nested maps by ignoring the nesting feature. For instance, in an *outline* view, nested maps might appear as follows:

```
Root
  Map
    Question
      Answer
        Pro Argument
        Con Argument
    Map
      Answer
        Pro Argument
```
Nodes Serialized But Not Included in Common Format
Suggested behavior: ignore those nodes.

Multiple Pages of Details from Compendium
Candidate suggested behaviors:
- Restrict Compendium to one page of details
- When multiple pages are detected:
  - concatinate and trim as needed

**ELEMENT = view** **ATTRIBUTES = Xposition and Yposition**
Suggested behaviors:
- Import: ignore unless its useful or can be transformed to local coordinates
  - Compendium will need to deal with auto arrange when it encounters "0" as values
- Export: default to "0"

**Simple Example XML Serialization from Compendium**
See **Appendix A** for the XML serialization.

A simple dialogue map with a nested map node was created to demonstrate a Compendium XML document. The full dialogue takes two images to display: Figure 1 and Figure 2.

![Figure 1: Outer (root) Map](image1)

![Figure 2: Nested Map](image2)

**Examining a Compendium Document**
The outer element is `<model rootview="...">`

A *model* is a container for the following:
- views
Models and Views and their Relation to Nodes

Note: Many of the attributes are slated to be removed from the Common Format DTD.

`viewref` refers to a view, synonymous with a map. `noderef` refers to a node identifier. Further explanation is necessary.

In Compendium, there is a Map node that is created in a project. That is the outer view. Its `viewref` points to a `noderef` that represents that outer map. That `noderef` becomes the `viewref` for the outer map such as Figure 1. The `noderefs` of that `viewref` will be the nodes contained in that view. In our example, one of those nodes is a map node. It becomes a new `viewref`, the `noderefs` for which are the nodes in the nested map, such as Figure 2.

Nodes and Details

Note that `details` are paged. In Compendium, there is the editorial option to create pages of details.

Links
<link id="19216811021244938254109" created="1244938254109"
    lastModified="1244938254109" author="jackpark"
    type="39"
    originalid=""
    from="19216811021244938215031" to="19216811021244938202218"
    label="" arrow="1">
<linkviews>
    <linkview id="19216811021244938183265"/>
</linkviews>
</link>
**Node Types**

From the Compendium source file IcompendiumConstants

```java
/**
 * This represents a map, which is a container for other nodes and links.
 * This can be used to:
 * - create a 'picture' of the relationships between ideas;
 * - group questions and ideas together in meaningful clusters;
 * - create associative links between nodes.
 */
public final static int MAPVIEW = 2;

/** This represents a Question or Issue for discussion. */
public final static int ISSUE = 3;

/** The represents an Answer or Position, often in response to a question or issue. */
public final static int POSITION = 4;

/** This represents a response in favour of an answer or position. */
public final static int PRO = 6;

/** This represents a response against an answer or position. */
public final static int CON = 7;

NOTE: In Compendium, there are occasions where a node is used more than once, prompting a *shortcut* type. A shortcut type is the node type + 10: a MAPVIEW shortcut = 12.
```
**Link Types**

From the Compendium source file IcompendiumConstants

```java
//LINK TYPES
/** Indicates the Node at the FROM end of the link responds to the Node at the TO end of the link */
public final static String RESPONDS_TO_LINK = "39";

/** Indicates the Node at the FROM end of the link supports the Node at the TO end of the link */
public final static String SUPPORTS_LINK = "40";

/** Indicates the Node at the FROM end of the link objects to the Node at the TO end of the link */
public final static String OBJECTS_TO_LINK = "41";

/** Indicates the Node at the FROM end of the link challenges the Node at the TO end of the link */
public final static String CHALLENGES_LINK = "42";

/** Indicates the Node at the FROM end of the link is a specialization of the Node at the TO end of the link */
public final static String SPECIALIZES_LINK = "43";

/** Indicates the Node at the FROM end of the link expands on the Node at the TO end of the link */
public final static String EXPANDS_ON_LINK = "44";

/** Indicates the Node at the FROM end of the link is related to the Node at the TO end of the link */
public final static String RELATED_TO_LINK = "45";

/** Indicates the Node at the FROM end of the link is about the Node at the TO end of the link */
public final static String ABOUT_LINK = "46";

/** Indicates the Node at the FROM end of the link resolves the Node at the TO end of the link */
public final static String RESOLVES_LINK = "47";
```
/** Indicates which link type is the default link type - Currently the 'Related To Link' */

public final static String DEFAULT_LINK = "45";
**DTD**

The following DTD is a subset of the Compendium DTD

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT model (views, nodes, links, codes, meetings*)>
<!ATTLIST model
    rootview CDATA #REQUIRED
>
<!ELEMENT page (#PCDATA)>
<!ATTLIST page
    nodeid CDATA #REQUIRED
    author CDATA #REQUIRED
    pageno CDATA #REQUIRED
    created CDATA #REQUIRED
    lastModified CDATA #REQUIRED
>
<!ELEMENT details (#PCDATA | page)*>
<!ELEMENT detail (#PCDATA)>
<!ELEMENT linkview (#PCDATA)>
<!ATTLIST linkview
    id CDATA #REQUIRED
>
<!ELEMENT linkviews (#PCDATA | linkview)*>
<!ELEMENT link (#PCDATA | linkviews)*>
<!ATTLIST link
    id CDATA #REQUIRED
    created CDATA #REQUIRED
    lastModified CDATA #REQUIRED
    author CDATA #REQUIRED
    type CDATA #REQUIRED
    originalid CDATA #REQUIRED
    to CDATA #REQUIRED
    from CDATA #REQUIRED
    label CDATA #IMPLIED
    arrow CDATA #REQUIRED
>
<!ELEMENT links (#PCDATA | link)*>
<!ELEMENT node (details*, detail*, source, image, background, coderefs, shortcutrefs, mediaindexes*)>
<!ATTLIST node
    id CDATA #REQUIRED
    type CDATA #REQUIRED
    extendedtype CDATA #REQUIRED
    originalid CDATA #REQUIRED
    author CDATA #REQUIRED
    created CDATA #REQUIRED
    lastModificationAuthor CDATA #IMPLIED
```

APPENDIX A—XML Serialization of TrivialMap Example

Note: some elements, e.g. `<code>` and `<coderef>` where hand removed as not applicable to the common DTD.

```xml
<model rootview="19216811011237140096791">
  <views>
    <view viewref="19216811011237140096791" noderef="19216811021244938183265" XPosition="201" YPosition="236" created="1244938183265" lastModified="1244938183265" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
    <view viewref="19216811021244938183265" noderef="19216811021244938215031" XPosition="207" YPosition="28" created="1244938215031" lastModified="1244938215031" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
    <view viewref="19216811021244938183265" noderef="19216811021244938202218" XPosition="16" YPosition="68" created="1244938202218" lastModified="1244938202218" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
    <view viewref="19216811021244938183265" noderef="19216811021244938232984" XPosition="316" YPosition="20" created="1244938232984" lastModified="1244938232984" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
    <view viewref="19216811021244938183265" noderef="19216811021244938256703" XPosition="177" YPosition="106" created="1244938256703" lastModified="1244938256703" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
    <view viewref="19216811021244938183265" noderef="19216811021244938304531" XPosition="16" YPosition="59" created="1244938304531" lastModified="1244938304531" showTags="true" showText="true" showTrans="true" showWeight="true" smallIcon="false" hideIcon="false" labelWrapWidth="25" fontsize="12" fontface="Dialog" fontstyle="0" foreground="-16777216" background="-1">
      ...
    </view>
  </views>
</model>
```
<node id="19216811011237140096791" type="2" extendedtype="" originalid=""
author="jackpark" created="1237140096775" lastModified="1237140096775" label="Home Window" state="2" lastModificationAuthor="jackpark">
<details>
<page nodeid="19216811011237140096791" author="jackpark" created="1237140096775" lastModified="1237140096775" pageno="1">Home Window of jackpark</page>
</details>
</node>

<node id="19216811021244938183265" type="2" extendedtype="" originalid=""
author="jackpark" created="1244938183265" lastModified="1244938190734" label="TrivialMap" state="2" lastModificationAuthor="jackpark">
<details>
<page nodeid="19216811021244938183265" author="jackpark" created="1244938183265" lastModified="1244938190734" pageno="1"></page>
</details>
</node>

<node id="19216811021244938215031" type="4" extendedtype="" originalid=""
author="jackpark" created="1244938215031" lastModified="1244938217953" label="Nothing" state="2" lastModificationAuthor="jackpark">
<details>
<page nodeid="19216811021244938215031" author="jackpark" created="1244938215031" lastModified="1244938217953" pageno="1"></page>
</details>
</node>

<node id="19216811011244938202218" type="3" extendedtype="" originalid=""
author="jackpark" created="1244938202218" lastModified="1244938211281" label="What shall we do next?" state="2" lastModificationAuthor="jackpark">
<details>
<page nodeid="19216811011244938202218" author="jackpark" created="1244938202218" lastModified="1244938211281" pageno="1"></page>
</details>
Disagree: we should be busy

An alternative idea

What about?

Let's write a story

Sounds good