

The Composite Gospel Index in RDF

This document describes a representation of the Composite Gospel Index (CGI) in the Resource Description Framework (RDF), an XML language for describing meta-data for web resources.

Table of contents

1 The Composite Gospel Index in RDF.....	2
1.1 Why RDF?.....	2
2 The Meta-data is the Message.....	2
3 The RDF Vocabulary.....	3
4 Developing Browsable Pericopes.....	3
5 Looking Ahead.....	4

Note:

You should start with [this overview](#) of the Composite Gospel Index if you're not already familiar with the concepts. You can find the data and applications themselves [here](#).

1 The Composite Gospel Index in RDF

The Composite Gospel Index (CGI) combines the four Gospel accounts of the life of Jesus into a single unified view. Instead of the traditional book/chapter/verse organization, it divides the texts into about 350 *pericopes* with a unique identifier and a brief descriptive label (e.g. #235, "Jesus speaks to a rich young ruler"). Each pericope is indexed to one or more Gospel passages that provide its source text.

At the time it was created, the CGI was the only XML-based composite of the life of Jesus that i had found (as of Dec 2004, this is still true). While an XML representation provides a well-defined and re-usable syntax, it's not as general as RDF for specifying relationships between data elements. So as i became more familiar with RDF, it seemed reasonable to convert to an RDF representation.

1.1 Why RDF?

A full examination of RDF is outside the scope of this page, but in brief, RDF is a language specifically designed for describing meta-data about resources. In the case of the CGI, the resources are the pericopes and their sources, and the meta-data include the Scripture references, the titles and identifying numbers, and their sequence information.

One key benefit of RDF is that it models relationships using a triples model (subject, predicate, object). The serialization of this model to XML tree-structured representation (properly called RDF/XML) is only one view. The triples model is quite general, and, used together with an ontology, allows a fuller specification of semantics than XML alone can provide. While the proponents of RDF also look forward to other benefits like automated discovery of meta-data, it's still a little early to see whether this will prove to be an important value or not. But RDF provides a standardized foundation for query and discovery that again goes beyond XML itself.


(A more comprehensive discussion of RDF can be found at [the W3C website](#).)

2 The Meta-data is the Message

Since RDF is a meta-data language, the normal function of an RDF description is to stand alongside an existing data resource. The Composite Gospel Index is a little different, since it is itself primarily meta-data: in other words, it is an organizational scheme tying together pericopes across the four Gospel sources, along with their various properties. The verse content of the sources provides value for human readers, but most of what's different about

the CGI is meta-data: grouping the sources and their references, providing descriptive titles, and sequencing, either by pericope sequence (the overall time sequence of life of Jesus), or by source sequence (how they are ordered by the original authors in their gospels).

As i became more familiar with RDF, i struggled to find the right application of it to the Composite Gospel. Ultimately i settled on two parallel versions, maintained programmatically from a single XML source. The "textualized" version has one file for each pericope, with the verse content inserted automatically from an OSIS-formatted Bible source. Though the file content is valid RDF (you can check that [here](#)), the files themselves are stored as XML so that browsers know how to display them (hopefully that will change over time as RDF becomes more mainstream). An XSL stylesheet is used to render the sources in columns, with hyperlinks for pericope sequence and source sequence. The primary purpose of these individual files is to support browsing applications. [Pericope 93](#), "Jesus describes his true family", is an example.

In addition to the textualized version, there is [a single RDF file](#)  with all the pericopes and no verse content. This more closely corresponds to a "traditional" use of RDF as a meta-data file, though in this case the resource it describes is really ... itself. The RDF file is created directly from the XML source (which is "[RDF friendly](#)"), using an XSL transformation.

3 The RDF Vocabulary

The vocabulary for the CGI in RDF is actually defined in [OWL](#), but since OWL is built on RDF, the vocabulary is properly also in RDF. The base of the ontology describes a vocabulary for textual elements (chapters, verses, and other kinds of passages, including pericope sources), as well as their relationships to other elements (the author, which book a passage is part of, and its verse content).

(Vocabulary description forthcoming ...)

4 Developing Browsable Pericopes

The "textualized" version is derived from an underlying XML representation. Using the PerlSAX parser, sequence relationships (`nextPericope`, `nextPericopeBySource`, and their inverses) are computed directly: that avoids manually specifying information which might have to change in the event of future modifications to the pericope sequence. The same is true for the starting chapter and verse for each `PericopeSource` (required for sorting the pericopes in the author's original order), and the number of verses in the verse content for each source (which also provides a handy measure of how much content each pericope includes).

The verse text comes from reading an OSIS-encoded XML file (also using PerlSAX) and synchronizing the content with the CGI structure. At present, the ESV is used: it's a modern version with an OSIS implementation.

An important future project will be to develop browsable pericopes for other languages than English. To do so requires only translating the pericope titles (355 short phrases) into the target language, and identifying an OSIS version of the Gospels in that language. Because RDF explicitly provides for specifying the language used in strings (using the `xml:lang` attribute), multiple translations of the pericope titles could be maintained within the same RDF structure. All the other triple information is language-independent. If you're able to help with a project like this, please [contact us](#).

5 Looking Ahead

In its current stage of development, the CGI is still simple enough that more complex representations like RDF aren't strictly necessary. But it's my intention to continue adding other meta-data, for example:

- organizing the pericopes by the larger historical periods of Jesus' life: for example, his early Galilean ministry, journey to Jerusalem, or Holy Week
- tying in the [New Testament Names](#), so you can retrieve the pericopes referring to specific individuals or places

RDF will make this expansion into additional meta-data feasible.