# Developing an XML-based, exploitable linguistic database of the Hebrew text of Gen. 1:1-2:3

Thesis by

**Jan Hendrik Kroeze**

**(21345393)**

Submitted in fulfilment of the requirements for the degree

**Philosophiae Doctor (Information Technology)**

in the

**School of Information Technology**

in the

**Faculty of Engineering, Built Environment and Information Technology**

**University of Pretoria**

**Pretoria**

**Promoter: Prof. Dr. T.J.D. Bothma**

**Co-promoter: Dr. M.C. Matthee** **April 2008**

# Developing an XML-based, exploitable linguistic database of the Hebrew text of Gen. 1:1-2:3

JH Kroeze

## Abstract

The thesis discusses a series of related techniques that prepare and transform raw linguistic data for advanced processing in order to unveil hidden grammatical patterns. A threedimensional array is identified as a suitable data structure to build a data cube to capture multidimensional linguistic data in a computer's temporary storage facility. It also enables online analytical processing, like slicing, to be executed on this data cube in order to reveal various subsets and presentations of the data. XML is investigated as a suitable mark-up language to permanently store such an exploitable databank of Biblical Hebrew linguistic data. This concept is illustrated by tagging a phonetic transcription of Genesis 1:1-2:3 on various linguistic levels and manipulating this databank. Transferring the data set between an XML file and a threedimensional array creates a stable environment allowing editing and advanced processing of the data in order to confirm existing knowledge or to mine for new, yet undiscovered, linguistic features. Two experiments are executed to demonstrate possible text-mining procedures. Finally, visualisation is discussed as a technique that enhances interaction between the human researcher and the computerised technologies supporting the process of knowledge creation. Although the data set is very small there are exciting indications that the compilation and analysis of aggregate linguistic data may assist linguists to perform rigorous research, for example regarding the definitions of semantic functions and the mapping of these functions onto the syntactic module.

# Summary

**Title:** Developing an XML-based, exploitable linguistic database of the Hebrew text of Gen. 1:1-2:3
**Candidate:** Jan Hendrik Kroeze (1958)
**Promoter:** Prof. dr. T.J.D. Bothma
**Co-promoter:** Dr. M.C. Matthee
**Departments:** Information Science; Informatics
**School:** School of Information Technology
**Degree:** Philosophiae Doctor (Information Technology)

The thesis touches on various sub-disciplines of computational linguistics, investigating the use of XML tagging to capture linguistic categories in the Hebrew text of Genesis 1:1-2:3 and to construct a threedimensional databank; using string processing algorithms to round-trip the data to and from the databank and computer program; using array processing to explore the semantic patterns hidden in the marked-up text; and using a graphical visualisation to investigate the mapping of semantic and syntactic functions. The thesis hopes to make a contribution by demonstrating the rigour enforced by the application of data-warehousing and data-mining concepts to a linguistic databank. It proposes a macro-structure that may be used in future to package and integrate multidimensional linguistic data.

Chapter 2 experiments with a threedimensional data structure, using Visual Basic 6, and finds that a threedimensional array could be used to represent inherently multidimensional linguistic data regarding Biblical Hebrew clauses. Various layers of linguistic knowledge can be integrated by stacking various modules of analysis onto each other.

Chapter 3 explores data warehousing and online analytical processing concepts to find ways to render meaningful subsets of linguistic data stored in a threedimensional array. Concepts like slicing and dicing are adjusted to make them useful for the processing of linguistic data.

Chapter 4 tries to find a more elegant solution for the permanent storage of the databank using XML technology. Due to its flexibility XML is chosen to build a text-based databank. The experiment indicates that XML is indeed a very suitable mark-up technology that can be used to permanently store the linguistic data in a separate databank because it allows users to create their own tag sets which may simulate a multidimensional database structure.

Chapter 5 investigates round-tripping in order to satisfy the requirement of finding a stable platform for the data, while also allowing editing and advanced processing of the data. In addition, various viewing and searching functions are discussed. Create, update and delete functionalities are added to enable users to populate and edit the clause cube while it is in the array state and to save these updates both to the RAM and on permanent storage in XML format.

Chapter 6 focuses on the benefits of text data mining facilitated by the preceding technologies. Some data mining concepts are applied in two experiments by aggregating aspects of the semantic and syntactic modules tagged in Genesis 1:1-2:3. Computer-assisted explorations of the semantic and syntactic data captured in the XML database illustrate the rigour enforced by such a text-mining venture.

In Chapter 7 projects are suggested (one of which is implemented) that could use the XML-based data cube of Genesis 1:1-2:3 in visualisation ventures to clearly show linguistic patterns uncovered by means of a computer program. These techniques may be used to create user-friendly interfaces that may facilitate easier and more intuitive mining of linguistic data.

**Keywords**

The following keywords represent the most important aspects covered in the thesis:

- Threedimensional array
- Online analytical processing (OLAP)
- XML

- Round-tripping
- Database management
- Data warehousing
- Text-data mining
- Computational linguistics
- Visualisation
- Hebrew Bible

# Acknowledgements

I would like to thank the following people:

- My supervisors, Prof. Theo Bothma and Dr. Machdel Matthee, for your excellent supervision and feedback on the related papers and articles produced during the execution of the doctoral project
- The Head and Acting Head of the Department of Informatics, Prof. Carina de Villiers and Prof. Trish Alexander, the Dean of the EBIT Faculty, Prof. Roelf Sandenbergh, as well as the Department's management committee, for granting me various shorter and longer periods of study and research leave to work on the research project, and for subsidising various overseas trips to read the papers that developed parallel to the chapters of this thesis
- Mr. Danie Malan of the University of Pretoria's library who helped me with literature searches and to find sources that were not always easy to get hold of
- All the editors and peer reviewers, as well as other colleagues, who assisted me with advice, help and comments that helped to steer the research in the right direction
- Ms. Mariëtte Postma for her excellent language editing
- My wife, Irma, for allowing me and assisting me to embark, for the second time, on the lengthy process of writing a doctoral thesis, and for helping to get aspects regarding the philosophy of science in the introduction right
- My son, Jan, for writing the Java progam discussed in Chapter 7 as an example of the graphical visualisation of linguistic data
- My daughter, Christien, for creating the beautiful graphics of threedimensional data cubes, used as illustrations in Chapters 2 and 3
- My parents, Jan and Clasie Kroeze, and my mother-in-law, Babs Jansen van Rensenburg, as well as my other family, in-laws and friends for your continued support and encouragement

# Declaration of originality

I declare that this thesis, *Developing an XML-based, exploitable linguistic database of the Hebrew text of Gen. 1:1-2:3*, is my own work, that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references, and that it has not been submitted for a degree at another university.

J.H. Kroeze

# Table of contents

**Chapter 1: Introduction** 1

1.1 Introduction 1

1.2 Background: forty years of Biblical Hebrew information systems 1

1.2.1 Levels of analysis 2

1.2.2 Underutilisation of existing tools 4

1.2.3 Integration as a solution to enhance utilisation 7

1.2.4 Visualisation and flexibility 9

1.3 Problem statement 11

1.4 Research questions 12

1.4.1 Main research question 12

1.4.2 Secondary research questions 13

1.5 Hypothesis 14

1.6 Positioning of Linguistic Information Systems within a research discipline 15

1.7 Research style and methodology 21

1.8 Research plan 23

1.9 Structure of the thesis 26

1.10 Contribution to the field of ICT 28

1.11 Definition of terms 29

1.12 Conclusion 31

**Chapter 2: Towards a multidimensional linguistic database of Biblical Hebrew** 33

2.1 Introduction 33

2.2 The need for integration 34

2.3 A clause cube as the ideal data structure 37

2.4 Implementing the clause cube in cyber space 47

2.5 Building and using a multidimensional database for Biblical Hebrew 54

2.6 Conclusion 55

**Chapter 3: Slicing and dicing the clause cube** 57
3.1 Introduction 57
3.2 Using a data cube to integrate complex sets of linguistic data 58
3.3 Processing the information in a clause cube 63
3.3.1 Rotation 64
3.3.2 Slicing 67
3.3.3 Dicing 72
3.3.4 Searching 73
3.4 Application: slicing and dicing the Genesis 1:1-2:3 clause cube 74
3.5 A comparison of data-cube concepts and clause-cube concepts 79
3.6 Conclusion 82

**Chapter 4: Building and displaying the clause cube using XML** 85
4.1 Introduction 85
4.2 Linguistic databases and computational linguistics 87
4.3 Linguistic layers 89
4.4 The phrase as basic building block of the database structure 90
4.5 Why should XML be explored as an option to build an exploitable database of linguistic data? 91
4.5.1 Why is XML suitable for implementing a database? 92
4.5.2 Why is XML suitable for linguistic data? 94
4.5.3 Why is XML suitable for data exploration? 97
4.5.4 What are the disadvantages of XML? 98
4.6 How can XML be used to build an exploitable linguistic data cube? 100
4.7 How can XML represent the syntactic and semantic analyses of free text? 101
4.8 How can XML represent inherently multidimensional data? 102
4.9 The structure of the Genesis 1:1-2:3 database in XML 103
4.10 Critical discussion of the XML clause cube implementation 109
4.11 Validating the XML document 113
4.12 Viewing the XML file in a web browser 119
4.13 Conclusion 122

**Chapter 5: Conversion of the Genesis 1:1-2:3 linguistic data between the XML database and the array in Visual Basic** 123
5.1 Introduction 123
5.2 Conversion between VB6 and XML (round-tripping) 128
5.2.1 From XML to VB6 130
5.2.2 From VB6 to XML 144
5.3 Editing the data in the clause cube 147
5.4 Conclusion 153

**Chapter 6: Advanced exploration of the clause cube** 155
6.1 Introduction 155
6.2 The need for advanced, computer-assisted exploration of linguistic data 156
6.3 Text mining as a knowledge-invention venture 159
6.4 Extraction and analysis of semantic role frameworks as a venture in text data mining 163
6.4.1 Slicing off the semantic functions layer from the data cube 164
6.4.2 Sorting the elements in each row 171
6.4.3 Concatenation of each row 175
6.4.4 Ordering the rows as units with reference to each other 179
6.4.5 Identifying and counting unique semantic frameworks 184
6.5 Analysis of the mapping of syntactic and semantic functions as another linguistic data-mining venture 193
6.6 Conclusion 242

**Chapter 7: Visualisation of the Biblical Hebrew linguistic data in the XML cube** 245
7.1 Introduction 245
7.2 What is visualisation? 247
7.3 Various approaches of visualisation 249
7.3.1 Text-based visualisation 249
7.3.2 Graphical visualisation tools 251
7.4 The purpose of visualisation 252
7.5 Requirements of visualisation tools 255

7.6 XML's suitability for visualisation 256
7.7 Some examples of visualisation of linguistic data 257
7.8 Application: a graphical topic map of semantic and syntactic mappings 262
7.9 Conclusion 269

**Chapter 8: Conclusion** 271
8.1 Introduction 271
8.2 Summary of thesis contents 272
8.3 Revisiting the research questions 276
8.4 Future research 279

**Bibliography** 283

**Addenda (files on CD)**

A **Addendum A: Linguistic data regarding Gen. 1:1-2:3 represented by a threedimensional array in Visual Basic 6** CD
AddendumA_DatabankModule_20080411_Fin.pdf
B **Addendum B: Source code of Chapter 3** CD
AddendumB_SourceCode_Chapter3_20080411_Fin.pdf
**Program:** Gen1Version15.exe
C **Addendum C: Phonetic transcription of the Hebrew alphabet** CD
AddendumC_PhoneticTranscription_20080411_Fin.pdf
D **Addendum D: Identification of phrase types** CD
AddendumD_WordGroups_20080411_Fin.pdf
E **Addendum E: Syntactic functions** CD
AddendumE_SyntacticFunctions_20080411_Fin.pdf
F **Addendum F: Semantic functions** CD
AddendumF_SemanticFunctions_20080411_Fin.pdf
G **Addendum G: Clause cube schema** CD
AddendumG_ClauseCubeSchema_20080411_Fin.pdf
**Schema:** Gen1_InputV15.xsd
H **Addendum H: XML clause cube** CD
AddendumH_XMLClauseCube_20080411_Fin.pdf

**Databank:** Gen1_InputV15.xml

I **Addendum I: XML clause cube corrected** CD

AddendumI_XMLClauseCube_Corrected_20080411_Fin.pdf

**Databank:** Gen1_InputV15b.xml

J **Addendum J: XML clause cube style sheet** CD

AddendumJ_XMLClauseCube_StyleSheet_20080411_Fin.pdf

**Style sheet:** Gen1XMLdb03c.css

K **Addendum K: Viewing the clause cube in a browser** CD

AddendumK_ViewingClauseCubeInBrowser_20080411_Fin.pdf

L **Addendum L: Source code of Chapter 5** CD

AddendumL_SourceCode_Chapter5_20080411_Fin.pdf

**Program:** Gen1_XML_VB6_CRUD_Beta15_Ch5.exe

**Databank:** Gen1_InputV15_RT1.xml

M **Addendum M: Source code of Chapter 6** CD

AddendumM_SourceCode_Chapter6_20080411_Fin.pdf

**Program:** Gen1_XML_VB6_CRUD_SemFSynF_Ch6.exe

**Databank:** Gen1_InputV15_RT1.xml

N **Addendum N: Visualisation program (graphical topic map)** CD

**Program:** semantics.bat

# List of Figures

**Figure 1.1.** An example of WIVU's morphological and syntactic information, which is interactively available on the web (www.th.vu.nl/~wiweb/const/index.htm). 6

**Figure 2.1.** An excerpt of the Lexham Hebrew-English Interlinear Bible (http://www.logos.com/products/details/2055). 35

**Figure 2.2.** An interlinear analysis of Jonah 1:1a in an HTML table format (Kroeze, 2002). 36

**Figure 2.3.** A threedimensional data structure that consists of a set of 27 sub-cubes arranged according to three rows, three columns and three depth layers. 38

**Figure 2.4.** The knowledge that is represented by a collection of interlinear tables, rendered threedimensionally as a clause cube consisting of layers of clauses and analyses stacked onto each other. 40

**Figure 2.5a.** Gen. 1:1a analysed according to phrases and linguistic levels. 42

**Figure 2.5b.** Gen. 1:4c analysed according to phrases and linguistic levels. 43

**Figure 2.5c.** Gen. 1:5a analysed according to phrases and linguistic levels. 44

**Figure 2.6.** A clause cube (orthographic view) containing real linguistic data of three BH clauses, Gen. 1:1a, 4c and 5a. 45

**Figure 2.7a.** Revealing data contained in the second slice of the cube by removing the top slice. 46

**Figure 2.7b.** Revealing data contained in the bottom slice of the cube by removing the top and middle slices. 47

**Figure 2.8.** A part of the code that creates a threedimensional array and populates it with the selected layers of linguistic data. 50

**Figure 2.9.** A representation of a hierarchical syntactic structure using various members of the same dimension and by allowing measures to occupy more than one cell of a member. 53

**Figure 3.1.** A series of twodimensional tables, each containing a multidimensional linguistic analysis of one clause. 59

**Figure 3.2**. A threedimensional clause cube. 62

**Figure 3.3.** Information revealed on the front side of the clause cube. 64

**Figure 3.4.** Information revealed on the top side of the clause cube. 65

**Figure 3.5.** Information revealed on the bottom side of the clause cube. 65

**Figure 3.6.** Information revealed on the right side of the clause cube. 66

**Figure 3.7.** Information revealed on the left side of the clause cube. 66

**Figure 3.8.** Information revealed by rotating the clause cube 180 degrees in a clockwise manner. 67

**Figure 3.9.** Information revealed by rotating the clause cube 180 degrees head over heels. 67

**Figure 3.10.** The top slice of the data cube, revealing the multi-modular analysis of the first clause. 68

**Figure 3.11.** The middle slice of the data cube, revealing the multi-modular analysis of the second clause. 69

**Figure 3.12.** The bottom slice of the data cube, revealing the multi-modular analysis of the third clause. 69

**Figure 3.13.** Information revealed by slicing off the first three planes from the front side of the clause cube. 70

**Figure 3.14.** Information revealed by slicing off the first four planes from the front side of the clause cube. 71

**Figure 3.15.** Information revealed by slicing off the first two planes from the front side of the clause cube. 71

**Figure 3.16.** Information revealed by slicing off the first plane from the front side of the clause cube. 71

**Figure 3.17.** A slice of the Genesis 1:1-2:3 clause cube that reveals the multi-modular analysis of Gen. 1:17a-18a (one clause spanning two verses). 76

**Figure 3.18.** The clause cube searched on a specific parameter. 77

**Figure 3.19.** Using the "Scroll through slice of syntactic frameworks" button. 78

**Figure 3.20.** Using the "Scroll through slice of semantic frameworks" button. 79

**Figure 4.1.** The hierarchy of the Genesis 1:1-2:3 clause cube as reflected by its XML implementation. 103

**Figure 4.2.** An example of an XML schema used to annotate text (Witt et al., 2005: 105). 105

**Figure 4.3.** The basic structure of the XML database of Genesis 1:1-2:3. 105

**Figure 4.4.** Two populated clause elements in the XML database. 107

**Figure 4.5.** The XML Schema used to validate the XML database of Genesis 1:1-2:3. 114

**Figure 4.6.** The XML style sheet used to display the XML clause cube as a series of twodimensional tables in the Firefox or Opera web browser. 119

**Figure 4.7.** The first two clauses of the XML clause cube as displayed in the Firefox web browser as two twodimensional tables. 121

**Figure 5.1.** An extract of the Genesis 1:1-2:3 XML clause cube, which is representative of the hierarchy and structure of the file. 125

**Figure 5.2.** VB6 code that could be used to create a threedimensional array and populate one clause element with several layers of linguistic data. 127

**Figure 5.3.** VB6 code used to convert linguistic data from XML format into a threedimensional array. 131

**Figure 5.4.** Example of VB6 code that could be used to validate syntactic function elements during the array state. 137

**Figure 5.5.** VB6 code used to display one clause's linguistic analysis in a series of textboxes and labels on the interface. 138

**Figure 5.6.** The end-result after converting data from the XML clause cube into a threedimensional array in VB6. 140

**Figure 5.7.** VB6 code used to scroll through the clause cube data. 140

**Figure 5.8.** VB6 code used to display a required clause using its array index. 141

**Figure 5.9.** VB6 code used to perform exact searches. 142

**Figure 5.10.** VB6 code used to perform searches on parts of strings. 143

**Figure 5.11.** VB6 code used to save clause cube data from the threedimensional array into permanent XML-formatted storage. 145

**Figure 5.12.** The VB6 code used to make space for a new clause record to precede the current one. 149

**Figure 5.13.** The VB6 code used to make space for a new clause record to follow the current one. 149

**Figure 5.14** The VB6 code used to save new or edited clause data to the 151

RAM.
**Figure 5.15.** The VB6 code used to delete a clause record. 152
**Figure 6.1.** The knowledge creation paradigm (adapted from Cannataro et al., 2002: 34). 161
**Figure 6.2.** The interface of the completed program with buttons added to direct the user to the data-mining experiments. 162
**Figure 6.3.** The interface of the experiment that analyses semantic frameworks in the clause cube. 163
**Figure 6.4.** VB6 code used to slice off the module representative of the semantic role frameworks, while validating and numbering the semantic functions. 164
**Figure 6.5.** VB6 code used to display the semantic functions slice in a textbox. 167
**Figure 6.6.** The semantic frames of the 108 extracted, validated and numbered clauses, shown in a textbox. 167
**Figure 6.7.** VB6 code used to sort the semantic functions in each row. 171
**Figure 6.8.** The logically-ordered semantic role frameworks of the 108 clauses. 172
**Figure 6.9.** Code used to concatenate the semantic functions of each clause into a unit. 175
**Figure 6.10.** Concatenated semantic role frameworks of the 108 clauses. 176
**Figure 6.11.** VB6 code used to order the rows, representing individual semantic role frames, as units with reference to each other. 179
**Figure 6.12.** Semantic role frames of all clauses ordered logically with regard to each other. 181
**Figure 6.13.** VB6 code used to identify and count unique semantic role frames. 184
**Figure 6.14.** The final results of the semantic role framework experiment. 185
**Figure 6.15.** The interface of the experiment that analyses the mapping of semantic functions on syntactic functions. 195
**Figure 6.16.** VB6 code used to validate and number the syntactic function slice. 195
**Figure 6.17.** The syntactic function slice represented by a logical numbering 197

system.

**Figure 6.18.** VB6 code used to validate and number the semantic function slice. 200

**Figure 6.19.** The semantic function slice represented by a logical numbering system. 202

**Figure 6.20.** VB6 code used to test all possible combinations of semantic and syntactic functions. 205

**Figure 6.21.** Initial, human-unfriendly, results of the algorithm that tests for all possible combinations of semantic and syntactic functions in the data set. 206

**Figure 6.22.** VB6 code used to build and display a human-readable list of combinations of semantic and syntactic functions. 207

**Figure 6.23.** Human-readable list of all possible combinations of syntactic and semantic mappings that may appear in Genesis 1:1-2:3. 211

**Figure 6.24.** VB6 code used to prepare and show human-readable output, listing all occurring mappings of semantic and syntactic functions and their frequencies. 223

**Figure 6.25.** Final results of the experiment that analyses the mapping of semantic and syntactic functions in Genesis 1:1-2:3. 224

**Figure 7.1.** A twodimensional visualisation of a sentence's analysis on various levels (Petersen, 2004b). 250

**Figure 7.2.** Proposal for a graphical-spatial topic map showing associations between some semantic and syntactic functions based on co-locations in phrases, based on an idea for literary analysis by Bradley (2003: 198). 259

**Figure 7.3.** A proposal for a graphical visualisation of a network of semantic frameworks in Genesis 1:1-2:3 containing purpose as one of their elements (based on an idea for lexical visualisation by Manning et al., 2001: 139). 261

**Figure 7.4.** Topic map of all phrases' syntactic and semantic functions as marked up in Genesis 1:1-2:3. 264

**Figure 7.5.** A textual representation of the phrases in the database, viewable in the visualisation program "semantics.bat". 265

**Figure 7.6.** Interface used to define and fine-tune filters in the visualisation tool. 266

**Figure 7.7.** A screen shot of a visualisation of the network linking the semantic function of product to the syntactic functions of either complement or object, as found in various clauses in the dataset. 267

**Figure 7.8.** Updated graph showing the network linking the semantic functions of product and patient to the syntactic functions of complement and object, as found in various clauses in the dataset. 268