Chapter 7
Visualisation of the Biblical Hebrew linguistic data in the XML cube[124]

7.1 Introduction

It is a known phenomenon that information technology changes the working culture and structure of organisations (see Du Plooy, 1998: 12-23). A similar phenomenon can be observed in other humanistic computing areas. For example, while electronic dictionaries initially mimicked printed lexicographical styles, with time researchers realised that the available technology actually enables many more ways to render these data. Slowly but surely this has lead to innovative ways of organising dictionary content (Manning et al., 2001: 136).

In a similar way, this chapter explores new renderings of linguistic data made possible by visualising XML-based clausal data. This is facilitated by the fact that the data in the underlying Genesis 1:1-2:3 threedimensional data cube is highly structured. Every phrase is tagged on various linguistic levels. The original text is thus not marked up only with inline elements[125] – every item is tagged for each level and all the data is stored in an XML file. If a phrase does not have a certain linguistic function, a null value or dash is inserted between the opening and closing tags. This approach is supported by XML's customisability and extensibility (Walsh & Muellner, 1999: 4).

[124] A paper based on this chapter ("Visualizing mappings of semantic and syntactic functions" by Kroeze, Bothma, Matthee & Kroeze, 2008) was read at the INFOS2008 conference. A second paper abstract that aims to implement other proposals made in this chapter ("Designing an interactive network graph of modular linguistic data in an XML database of Biblical Hebrew" by Kroeze, Bothma, Matthee, Kroeze & Kruger, 2008) has been accepted by the AIBI-8 conference, but should be regarded as a post-doctoral project.

[125] With inline tagging "you choose to mark up not every possible item, but only those for which distinctive tagging will be useful in the production of the finished document for the readers who will search through it" (Walsh & Muellner, 1999: 37).

The mark-up provides the semantics that facilitates not only sharing, exchange and manipulation of data (Rob & Coronel, 2007: 576), but also exploration of it. The core of information visualisation is indeed to allow "people at all levels of an organisation to converse with their data and, from these conversations, glean the patterns and trends that will help them become more efficient, productive and successful" (Freedman, 2006: 25). Visualisation should also enable computational linguists to converse with their data in such innovative ways.

According to Manning et al. (2001: 136) there have, however, been surprisingly few attempts to use visualisation techniques to enhance the use of electronic dictionaries. "Despite decades of highly creative and sophisticated innovation, and a plethora of claims for obvious superiority of the visualization approach, we do not see visual maps of verbal information in popular and effective use" (Landauer et al., 2004: 5219). The same situation is probably still true of other ventures in electronic linguistics projects. This chapter attempts to make a small contribution to fulfil this need.

The chapter will first give a general idea of the concept of visualisation as a venture in humanities computing before moving on to discuss various approaches and goals pursued by it. After an overview of the requirements that visualisation tools have to fulfil, XML is identified as a suitable technology for the capturing of data underlying the visualisation ventures. Some examples of lexicographical and literary visualisation attempts are discussed, and proposals are formulated regarding the adaptation of these visualisation approaches for other types of linguistic data. Finally, a graphical application on the Genesis 1:1-2:3 clause cube is discussed to practically illustrate some of these ideas. The text-mining process, facilitated by the graphical topic map tool, may be regarded as a form of linguistic, multidimensional online analytical processing.

7.2 What is visualisation?

Visualisation is a graphical display of data that facilitates information retrieval and exploration. It is an interdisciplinary activity that has links to the information and communication technologies of Information Science, Information Systems and Computer Science (cf. Eden, 2005: 61-62). Visualisation is used in advanced information retrieval techniques. The selection and manipulation of the subset of the dataset that will be displayed are based on algorithms that are researched by computer scientists. Computer Science makes another contribution in terms of the programming of graphical interfaces. This chapter, however, will concentrate on the ties between visualisation and databases, building on the idea of the use of XML to develop an exploitable database of linguistic data. The underlying data to be visualised should be stored in some or other databank, such as a relational database[126], XML file or multidimensional array (cf. Chapters 2 and 4). One has to remember that a lot of theory has already been encoded into the structure of the databank and that its use will be restricted to these confines (Neyt, 2006: 8).[127]

Using visualisation techniques in a project like this is a way of adopting a more holistic approach that is in line with an externalist view[128] of good science, which approves the incorporation of insights from other disciplines, especially in a diverse discipline like Informatics (Dennis et al., 2006: 7-8). Another link between Informatics and visualisation is the study of Human Computer Interaction, which is used to evaluate visualisation products and to test their usability.

A graphical visualisation tool uses all of these underlying technologies to present the selected data as a picture. This facilitates the exploration of the data, preferably by

---

126 "Humanists have used relational databases as the engines behind complex visualization systems, text archives, and multimedia works" (Ramsay, *s.a.*).

127 In this project these assumptions are encoded in the names and definitions of word groups, syntactic and semantic roles. These are based largely on the insights of SC Dik's (1997a; 1997b) Functional Grammar, especially in the case of semantic functions. See Addenda D – F.

128 An internalist view, on the other hand, argues "that a core set of knowledge and shared scientific paradigms generated internal [*sic*] to the discipline are hallmarks of mature science, and thus diversity is to be avoided" (Dennis et al., 2006: 7).

providing an interactive modus operandi. It therefore comes as no surprise that various authors refer to the data-mining operations made possible by visualisation tools. According to Keller et al. (2006: 44-45) information visualisation is the interactive, graphical rendering of abstract data to enhance information retrieval, data mining and learning. Many data-mining ventures start with a "hunch", a nagging feeling that there just might be an interesting relation between some of the elements in a dataset. Visualisation is a way to make explicit these beliefs and assumptions of a researcher, a way of "organizing information so as to facilitate making the recommended inferences" (Unsworth, 2001).

The relationship between data mining and visualisation is reciprocal. Data mining may be used to facilitate visualisation, and visualisation may be used to do interactive data mining. Interactive data mining requires cooperation between the database management system, the data-mining tool and the visualisation tool. The following scenarios are possible (Thuraisingham, 2002: 87-89, 279):

- After using a data-mining tool to identify patterns in the data the results are presented in a user-friendly interface using visualisation techniques.[129]
- Essential elements of the raw data are first represented visually and then explored using data-mining algorithms.
- Visualisation techniques are used to complement data mining, e.g. to fine-tune or better understand correlations and patterns already found.
- Visualisation steers data mining, for example to dynamically change the focus of the mining process or to implement what-if scenarios.

To conclude this section, a final definition of visualisation may be formulated as follows: visualisation is a graphical display of subsets of a dataset, based on attributes that are linked by means of keys, array indexes or mark-up tags in order to facilitate a preferably interactive exploration of the data.

[129] According to Freedman (2006: 24) information visualisation is "an interface that sits atop the data mining reporting program" bringing to the front complex patterns hidden in multidimensional data.

7.3 Various approaches of visualisation

Visualisation may be regarded as the third step of computerised text analysis. After an archive or database has been built during the initial meta-linguistic phase to create a marked-up version of a literary text, software is developed in the algorithmic phase to analyse the source materials. These phases are followed by the representational phrase, which presents the interpreted data in a way that satisfies the needs of the user (cf. Neyt, 2006: 2-5). In more advanced approaches visualisation may also be used to facilitate data exploration (see the section on the purpose of visualisation below).

7.3.1 Text-based visualisation

The various approaches towards visualisation may be categorised according to the number of dimensions represented. A text-based representation of the results of a data-mining exercise may be regarded as a onedimensional rendering of the data. It is a stream-based, non-graphical, presentation that only qualifies for the term visualisation in the widest sense of the word. The results of the analyses of semantic frames presented in a textbox may be regarded as such a simple onedimensional display of the results of a rather advanced algorithmic phase (see Figures 6.14 and 6.25). It requires the user to read linearly in order to access the information contained in it. Various fonts, colours and backgrounds could be used to highlight the various linguistic layers to improve the usability of the interface (cf. Neyt, 2006: 2-7).

Another, more advanced, text-based visualisation tool could be built using some of the guidelines given by Sinclair (2003: 178-180). His point of departure is to conserve the original text as a basic interface in order not to alienate users who do not have advanced computer-literacy skills. Because literary critics are most familiar with printed texts, a visualisation tool should not banish it, but rather exploit it as a user-friendly angle of incidence. This can be implemented by showing "the text as a readable whole" on the screen Sinclair (2003: 178). Although a search function is very simple it still is one of the most powerful functionalities of electronic text: the

user could, for example, click on a word in order to jump to the next instance of the same word. The frequencies of word occurrences could be indicated by various shades of a specific colour. Pop-up boxes could be used to unveil related information when the user lets the mouse hover over a specific word or phrase. Words could also be used as multidimensional links by using menu options to trigger available analysis options for selected words.

A table is a relatively simple, but very efficient, twodimensional, non-graphical rendering of data. A clause can be analysed very effectively using columns to represent words and rows to show the various linguistic modules. Petersen (2004b) analyses the sentence "The door was blue" on the levels of word, phrase and clause, using a table with five columns and four rows (see Figure 7.1).[130] A similar approach is followed by Kroeze (2002) using linguistic data of Jonah. The cells on the various levels could "span" different parts of the primary data (original text) to reflect the unique partitioning on the different layers (Witt, 2005: 76-77). However, spanning is difficult to implement in a data cube, and will not be discussed in further detail in this study.[131]

	1	2	3	4
Word	w: 10001 surface: The psp: article	w: 10002 surface: door psp: noun	w: 10003 surface: was psp: verb	w: 10004 surface: blue psp: adjective
Phrase	p: 10005 phr_type: NP		p. 10006 phr_type: VP	p. 10007 phr_type: AP
Clause	c: 10008			

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

[130] Compare Petersen (1999: 14) for a more extensive example including a relative clause.

[131] Compare Witt (2005) who suggests that overlapping structures be annotated in separate XML documents using heterogeneous tag sets but linked by using the basic text as primary data and implicit links. Other solutions for this complex problem are CONCUR (available in SGML but not in XML), milestone elements ("empty elements which mark the boundaries between elements, in a non-nesting structure"), fragmentation and nesting, virtual joins, redundant encoding and standoff annotation (Witt, 2005: 58-59).

A style sheet, such as a css file, used to show the contents of the XML clause cube as a series of interlinear tables in a web browser, is a mechanism to implement a twodimensional representation of the XML data cube.[132] To show all of the data, a series of disconnected tables have to be used. Although a table uses two dimensions it is essentially still text-based and does not utilise all of the available ICT technologies discussed above.

7.3.2 Graphical visualisation tools

A graphical visualisation tool, however, uses more of the related technologies to present the data, for example, as a graph of connected nodes. The relations are based on data attributes that are linked by means of keys, array indexes or mark-up tags. The nodes and links form a picture that visually represents the interrelated data attributes. Other types of graphical visualisations are animation (Flash presentation), visualisation of a DTD as a tree structure, and visualisation of an archive as a lattice (Unsworth, 2001). These graphical visualisations could still be twodimensional, but also threedimensional or multidimensional. Although a computer screen is, like paper, essentially a twodimensional medium, it can be used inventively to simulate threedimensional models. Ideally, one should explore the possibility of three- or multidimensional visualisations to render inherently multidimensional data, such as linguistic analyses. "It is our conjecture that linguistic meaning is intrinsically and irreducibly very high dimensional" (Landauer et al., 2004: 5214). This is especially the case for the Genesis 1:1-2:3 data that has already been captured in a threedimensional data structure.

While the twodimensional table approach, discussed above, is very limited and cannot visualise a collection of clauses in one "picture", a threedimensional picture could show the various sentences stacked as layers in a cube. This would be very

[132] Internet Explorer 6 was not able to use the style sheet, created in Chapter 4, to directly show the XML data in table format and Opera7 was used instead. An alternative approach could be to use a detour by means of XML data binding in MSIE 5.0. A separate HTML file is created that contains code to incorporate the XML data and bind it to an HTML table (Rob & Coronel, 2007: 582-583).

difficult or almost impossible to represent on paper, but computer animation, on the other hand, can facilitate threedimensional simulation far better because "it can render movement and time" (Neyt, 2006: 7). Movement could be used in rotation to reveal the various faces of the threedimensional clause cube, as well as in slicing and dicing and drilling-down processes.

Although a multidimensional approach could be a better approach, it is not necessarily always the case. One should remember that readers are more used to twodimensional representations, which are also easier and less expensive to build (Eden, 2005: 62). Keller et al. (2006) found that, although twodimensional representations and the use of colour coding indeed enhance data mining and learning in comparison to pure text-based renderings, multidimensional approaches lead to cognitive overload on the user, which nullifies any additional benefits. However, they leave room for threedimensional visualisation of datasets where integration is important: "...threedimensional displays are superior to twodimensional ones only for specific tasks requiring integrating information over three dimensions" (Keller et al., 2006: 49). Since the Genesis 1:1-2:3 clause cube does integrate various linguistic levels (e.g. morpho-syntax, syntax and semantics), a threedimensional visualisation should be a viable option.

An interlinear twodimensional approach would probably be sufficient for a Bible reader or user who only needs enough information to understand the Hebrew text. For the in-depth researcher, however, threedimensional displays could be very helpful, if it could reflect the complex, underlying, multidimensional patterns and structures, built covertly into a stream of onedimensional text. However, this thesis will only discuss twodimensional graphs as a data-mining utility in detail.

7.4 The purpose of visualisation

Classic information retrieval is search-dominated and only effective if the user already has a good idea about the problem space and knows what to look for. "Such interfaces are ineffective for information needs such as exploring a concept"

(Manning et al. 2001: 136). Visualisation provides an alternative that could fill this gap.

Visualisation is an innovative way of representing data, one of the basic ventures of humanities computing (Neyt, 2006: 2-5). "Information visualization can present multiple dimensions of information that can be extremely useful in helping an analyst quickly sift through information to find patterns, filter the data live, and drill down to more meaningful result sets. These result sets can subsequently be exported via XML to other analytical packages" (Freedman, 2006: 24).

According to Freedman (2006: 24), the visual presentation of data is necessary

- to help the vast majority of visual learners to easier pinpoint the suspected correlation in data
- to make it easier to see relationships in large and complex sets of data
- to facilitate exploration of data to discover unexpected patterns

Like other text-analysis tools visualisation tools can simply be used as an interface both to find evidence to verify or falsify a theory (Rockwell, 2003: 217). Ideally, a visualisation tool should allow interactive operations so that the user can try out various scenarios and make adjustments to change or refine questions. Such an iterative process provides an experimental, almost "playful", way to do data mining in texts and this helps the researcher to question and even circumvent stereotyped hypotheses. Although not all results will be useful, this trial and error process could lead to the discovery of new, coherent patterns which would not be suggested by existing theory (Rockwell, 2003: 211-213).

Because play "thrives on improvisation and imagination", such a tentative, investigational research method complements the more rigorous and purposive research activities such as hypothesis testing (Sinclair, 2003: 181). Experimental exploration may stimulate new ideas and may be used to formulate new hypotheses, which may then be examined in more traditional, empirical ways. Sinclair therefore advocates "a hybrid model of formal, analytical functions and interpretive, exploratory

functions".[133] In this way, the visualisation tool may be used to support the formulation of a mental model (Bradley, 2003: 185). "It is, of course, possible to go to a literary text armed with a hypothesis, but we do better to go to it with a hunch borne of our collective musings – a sneaking suspicion that looking at it *this* way will turn up something interesting. Or better still, we could go to it with a machine that is ready to reorganize that text in a thousand different ways instantly" (Ramsay, 2003: 171).

According to Sinclair (2003: 182) computer-assisted "play" or experimentation is a suitable method for humanistic computing of literary texts, and visualisation may be used to implement this. Computer-assisted reading and text synthesis are other ways of making the use of computers more acceptable to humanistic scholars. Computer-assisted reading could be regarded as a more advanced visualisation technique than an interlinear approach, which is, however, still text-based. Computer-assisted text synthesis could be used in the report function of a visualisation tool to produce suitable outputs of research results, to reproduce the original text (without mark-up and analysis), or to create an amended text according to the end-users' requirements.

Besides its obvious applications for analysis by the intelligence community and for knowledge management in businesses information, visualisation may also be used for "exotic applications" by genealogists, lawyers and museums (Freedman, 2006: 25). If humanistic computing qualifies for the "exotic-application" tag, linguists may also use visualisations to highlight hierarchies, taxonomies and correlations in their datasets. Text analysis may be regarded as a balancing act between formal and interpretive tasks. An algorithm performing analytic functions on language may be regarded as a tool that takes responsibility for the more formal tasks and frees the hands of the human analyst to focus on the more non-deterministic activities (Bradley, 2003: 185).

Using visualisation techniques for linguistic software may also meet the educational needs of users who are not highly trained linguists, for example dictionary users. By

[133] According to Keller et al. (2006: 46) a combination of text and graphics are ideal for information processing since it uses both the verbal and visual memory systems.

providing affordable, customisable access to linguistic sources, it may help to keep endangered languages alive (Manning et al., 2001: 135, 137). Biblical Hebrew scholars should also make use of these technologies to protect and promote knowledge and research of this ancient language. According to Andersen & Forbes (2003: 44) one of the requirements of a proper rendering of syntactic structures of BH is that it should be pictorial, that is "clearly and concisely diagrammed". This chapter applies this principle to various linguistic modules (e.g. syntax and structural semantics) and trust that the proposal will make a contribution to the field of Biblical Hebrew linguistic information systems by facilitating cross-modular research.

7.5 Requirements of visualisation tools

The characteristics of a tool should differ according to the purpose, target audience and education level of the users. If an interactive interface is built for unsophisticated users, too much detail could lead to confusion and it would be better to use a simple and clean graphical layout (Manning et al., 2001: 138). This could be a valid requirement even if the users do have a lot of knowledge regarding the underlying linguistic data but not about computing, as is often the case in the humanities.

The interface should also be user-friendly, for example by providing meaningful and readable labels. It should allow end-users to visually rearrange the data to create suitable information (cf. Eden, 2005: 61-63). The analyst must be able to refine his/her query to more sharply focus on an uncovered pattern in order to better understand the relationship. Such an easy to use interface could help to involve more people "to take an active role in data mining activities" (Freedman, 2006: 24-25).

Furthermore, a visualisation tool should allow the user to adapt queries in an interactive way by dynamically mapping the underlying data and the resulting graphs in real time (Freedman, 2006: 24-25). This requires the underlying database to be integrated with the GUI. Although more than one visualisation or even a customisable visualisation would facilitate more interpretations, one has to acknowledge that even

these are limited to the theory and interpretation of the persons who tagged the data and created the software to access and analyse it (cf. Neyt, 2006: 7).

A visualisation tool should also allow scalability. The user should be able to work with anything from small sets of static data to large sets of changing data (Eden, 2005: 61). He/she should be able to adjust the granularity of reports because "too much information can cause the screen to resemble a giant hairball". Users with different needs should be able to either request high-level, global overviews or to drill down to the low-level, nitty-gritty details. The tool should also be able to visualise the results of both qualitative[134] and quantitative investigations (Freedman, 2006: 25).

Text analysis is not only a process of analytic decomposition, but also recomposition, both of which phases take place in a circular or spiral-like fashion (cf. Sinclair, 2003: 181). A complete tool should therefore also address the synthetic processes in reproducing the original text or creating new texts such as annotated versions or reports. The reporting module should include facilities to efficiently and easily communicate findings to other persons concerned (Freedman, 2006: 24). The reports should be customisable so that it can be adjusted for different audiences. A onedimensional text-based version should be provided as an alternative for non-visually oriented users (cf. Eden, 2005: 61-63).

7.6 XML's suitability for visualisation

An XML file itself is not a user-friendly document to read. Even though it is text-based and does not use complex programming syntax, the tags obscure the underlying text. The reader has to understand the basic hierarchy and schema to make any sense of the text file. Even if one knows the structure, it is not nearly as easy to read as a database table. XML is essentially a onedimensional stream of text. Even the use of indentation to highlight the hierarchy of tags is "just barely two dimensional" (Bradley, 2003: 199). However, many features may be recorded as mark-up that may be used by other programs to produce suitable visualisations. XML's weakness is therefore

[134] The visualisation of qualitative data is one of the challenges for software creators (Eden, 2005: 60).

also its strength: there exists one master copy of the marked-up text, which can be formatted in many different ways according to the needs and wishes of researchers and users (Flynn, 2002: 57).

Although an XML file uses relatively little space, due to the fact that it is text-based, it could still become too big. When it becomes so large that it slows down the parsing speed to an unacceptable level or that the data becomes too dense to be shown on a computer screen, the parsing algorithm should be adjusted to parse subsections of the database as required (cf. Manning et al., 2001: 141). The XML database may be broken down into logical and physical chunks so that different books or sections may be stored in separate files, which are defined as entities in the original document and inserted into it when referenced (cf. Walsh & Muellner, 1999: 30).

While the size of the XML databank was not a problem in the limited experiments of this thesis, the problem of handling embedded clauses could complicate matters. The problem was handled by presenting these clauses as separate entries linked by means of the clause IDs. This solution is similar to the approach followed by Manning et al. (2001: 141) to handle subentries in a dictionary. It facilitates not only a simpler XML structure but also easier visualisation strategies.

7.7 Some examples of visualisation of linguistic data

In this section, a number of linguistic visualisation projects will be discussed and suggestions will be made on how to adapt their approaches for the case in hand.

Bradley (2003:197-199) discusses feature structures and topic maps as examples of electronic tools that support the creation of mental models regarding literary analysis. A feature structure is a logical organisation of concepts identified in a text. The concepts are marked in the original texts by means of tags. Related concepts are linked in an intermediary document and these are then reorganised to form a final conceptual model as output. This could be visualised as a mind map, which is still more logically than graphically oriented. In a grammatical project a similar approach

could be used, for example, to compile a concept map of semantic functions and their relations to the various predication types.

A topic map contains a spatial element and therefore is more suitable for graphical visualisation. The researcher, for example, identifies various topics in a series of literary texts and draws a picture with the help of a visualisation tool linking these topics to the texts where they appear. Associations between the topics are also shown. In a grammatical project (such as the Genesis 1:1-2:3 clause cube) a topic map could be used to indicate the associations between semantic and syntactic functions. The mapping of semantic functions onto syntactic functions forms a complex network of associations in a text. A traditional interlinear paper-based analysis cannot show this network. A visualisation tool could make these associations visible in a similar way that it can give a better understanding of the semantic networks in a dictionary (cf. Manning et al., 2001: 137). In Figure 7.2 a tentative topic map of some semantic and syntactic functions is proposed as an example.

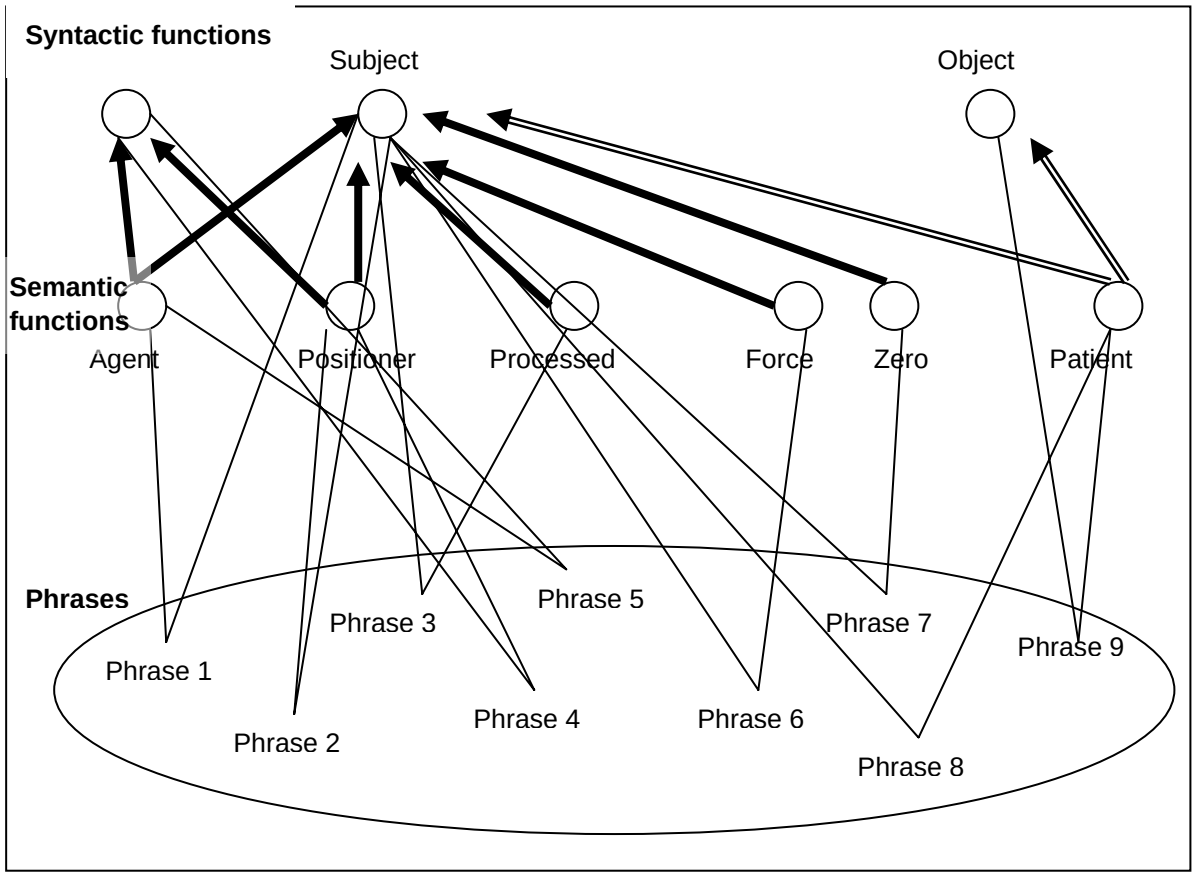

Figure 7.2. Proposal for a graphical-spatial topic map showing associations between some semantic and syntactic functions based on co-locations in phrases, based on an idea for literary analysis by Bradley (2003: 198).

In the proposed topic map, concepts (the semantic and syntactic functions) are represented as nodes in a twodimensional picture. Arrows are used to indicate the mapping of semantic functions onto syntactic functions, for example, agent, positioner, processed, force and zero are all first arguments, expressed by subjects in the surface structure of clauses. In passive clauses, agent and positioner may be expressed as adjuncts on the syntactic level. Heavy-weighted, single-line arrows flowing from the underlying semantic functions towards the syntactic functions mark these first arguments. Similarly, patient is a second argument in the logical structure, which may be expressed either by a (direct) object in an active realisation, or by a subject in a passive realisation. Double lines flowing from the semantic functions towards the syntactic functions mark it. Satellites could be indicated by arrows with three lines. If one would like to use the data in the Genesis 1:1-2:3 clause cube, it would not be possible to differentiate between first and second arguments and

satellites since this detailed information has not been tagged. Instead, a single line may be used to link any type of semantic function to the syntactic function realising it. In order to facilitate a drill-down facility, single lines are used in Figure 1 to connect both semantic and syntactic functions with the phrases in the dataset where they are mapped onto each other. As is already evident from the tangled web of lines in this rather simple proposal, the researcher will need a filtering facility to hide unnecessary information. This proposal has been applied in a Java project (see 7.8 below).

A graphical alternative to a topic map is a network graph. Manning et al. (2001: 139) give an example of a graphical rendering of related lexemes in a dictionary. These words form a graph with nodes and links. In this project a similar approach may be used to visualise the frameworks of semantic functions. It could, for example, show all the semantic frames in Genesis 1:1-2:3 containing purpose as one of its elements as proposed in Figure 7.3. The frames are linked by double lines connecting identical predicate types (action, position, process or state) and by single lines, based on shared semantic functions. The semantic function that is selected as focus point (for example by clicking on it in a drop-down list) is highlighted.[135] This brings together all the essential information about the semantic frames in which purpose appears and may help the researcher to either confirm his/her theory or to falsify it, or to reveal new patterns and prompt new hypotheses (for example, that the semantic function of purpose does not only occur after controlled predicates, but also in states). If only a formatted text document is available, the researcher first has to search for all frames containing purpose and then analyse this by hand to reach the same point. More colour could be used by highlighting the various semantic functions and predication types in different colours and by using different shades of one colour to indicate the frequencies of each frame. A drop-down list could be added to provide hyperlinks to the clause instances of each frame. Like a drop-down list showing an alphabetical list of lexemes in a dictionary this may provide "concreteness" or "tangibility" to the interface by providing access for the user to "what lies among the electrons beyond the screen" (ibid.). When the user sets the focus on another semantic function, the graphical interface should change accordingly. One or more multi-line textbox panes

[135] This may be regarded as a "focus + context strategy, which shows details at a focus point chosen by the user while still keeping the context or overall overview" as opposed to an "overview + detail strategy" that works the other way round (Eden, 2005: 63).

may be added to show more detail on the frames or functions, for example a relevant extract of a text file (cf. ibid.). A spring algorithm may be used to enable the user to interactively move the nodes on the screen – the algorithm should be intelligent enough to readjust the information accordingly. Such an "opportunistic exploration of networks" could support the researcher's data mining activities by revealing new patterns (see ibid.: 142, 147).

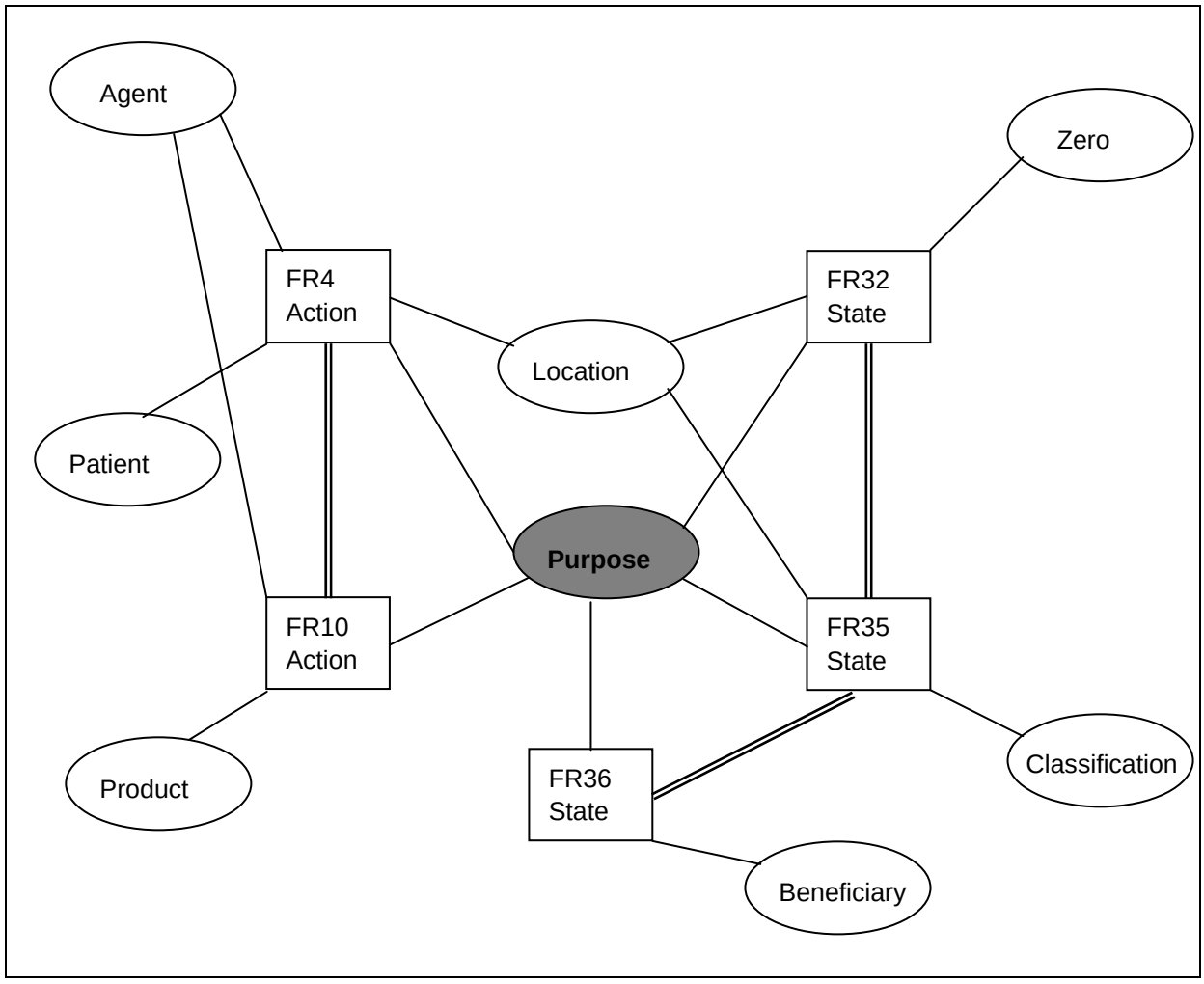

Figure 7.3. A proposal for a graphical visualisation of a network of semantic frameworks in Genesis 1:1-2:3 containing purpose as one of their elements (based on an idea for lexical visualisation by Manning et al., 2001: 139).

In Figure 7.3 above, the researcher wants to study semantic frames containing the semantic function of purpose. When he/she selects this function, the node is highlighted, and lines indicate the frames in which it appears within the dataset.

Purpose occurs in five frames. Each frame is again linked to all other semantic functions occurring in them. Frame 10, for example, consists of an action predication, in combination with agent, product and purpose elements. Purpose also appears in frames 4, 32, 35 and 36 (cf. Figure 6.14). This information may be used, for example, to check definitions of semantic functions. In this example a filter should have been used to show only those networks that contain purpose in their frameworks. Such a graphical tool could provide a more interactive, visual way to get to the same results reached in Chapter 6 (6.4.5).

More graphical possibilities that may be researched, are:

- Indication of frequencies by the use of colours or by adjusting the weights of the connecting lines
- Drilling-down facilities, for example, a right-click activated procedure that shows a pop-up window listing relevant examples
- Moving between various graphical implementations such as a threedimensional version of the clause cube and twodimensional slices
- Interactive, graphical rotation, slicing and drilling down of the clause cube
- Interactive focussing on more interesting frames; for example, since frames with a low frequency could offer more interesting or less generally known information, the researcher should be able to prioritise identified networks

In the next section one example of an adjustable graph will be discussed in detail to illustrate some of the concepts and benefits identified above. Due to the fact that the possibilities are more or less unlimited, many of the attractive ideas mentioned above will be left as opportunities for future research.

7.8 Application: a graphical topic map of semantic and syntactic mappings

In Chapter 6.5 the mapping of semantic roles on syntactic functions in the dataset was studied by creating a data-mining program that presented the results in a textual format. The same exercise is repeated here, but using a graphical tool that visualises

the same data. It also allows the researcher to experiment in a trial and error way by adding, removing and moving various filters in order to focus on required aspects.

An interactive visualising tool enables the researcher to look at a dataset from various perspectives. An example of such a tool is a graphical topic map[136] that shows all the relationships between phrases, semantic roles and syntactic functions in the dataset. A new program, using the same XML dataset, was created to facilitate link analysis between these nodes.[137] When one opens the program "semantics.bat" (see Addendum N)[138] the data file that has been used in the previous session is opened. One may click on the "File" menu to browse for and open the required file. In this case, the XML database, used in the previous two chapters is selected (Gen1_InputV15_RT1.xml). All the phrases in the database are shown with links to their semantic and syntactic functions. The semantic and syntactic functions that are mapped onto each other, are also linked. The data is still unfiltered and, therefore, looks like a hodgepodge of links (see Figure 7.4). All the semantic functions appearing in Genesis 1:1-2:3 are shown in the upper block; the syntactic functions are displayed in the middle-block and the phrases in the lower block.

136 Compare Bradley (2003).

137 This tool was created by Mr. J.C.W. Kroeze, a B.IT student at the University of Pretoria. He used the candidates's XML databank and designed the topic-map tool according to the candidate's specifications and parameters, more or less as suggested above (see, especially, Figure 7.2).

138 The applicable version of the *Java Runtime Environment* should be installed on a computer in order to run this program (see http://www.java.com/en/).

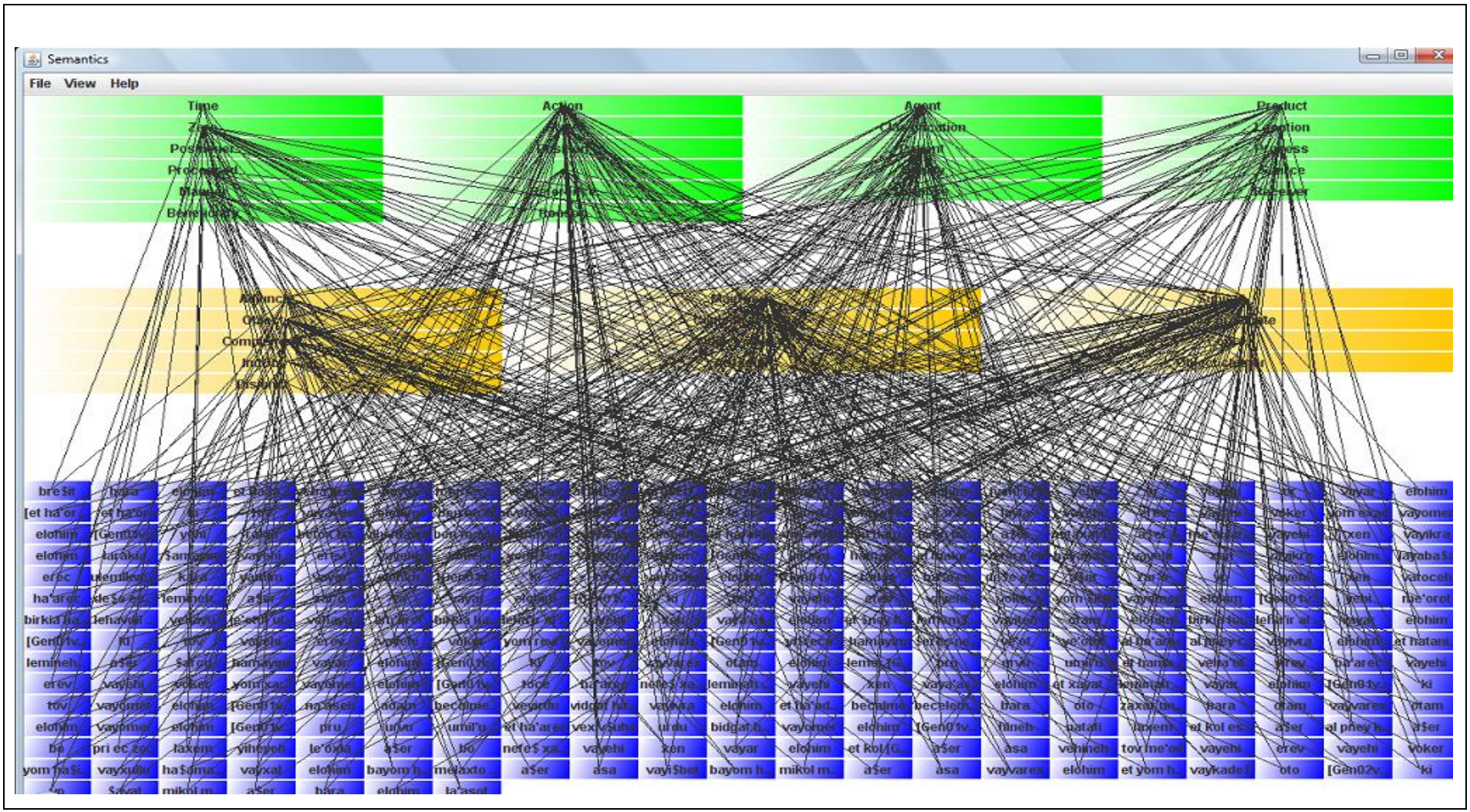

Figure 7.4. Topic map of all phrases' syntactic and semantic functions as marked up in Genesis 1:1-2:3.

The "View" menu allows the researcher to view the constituents' data in a textual format (see Figure 7.5).

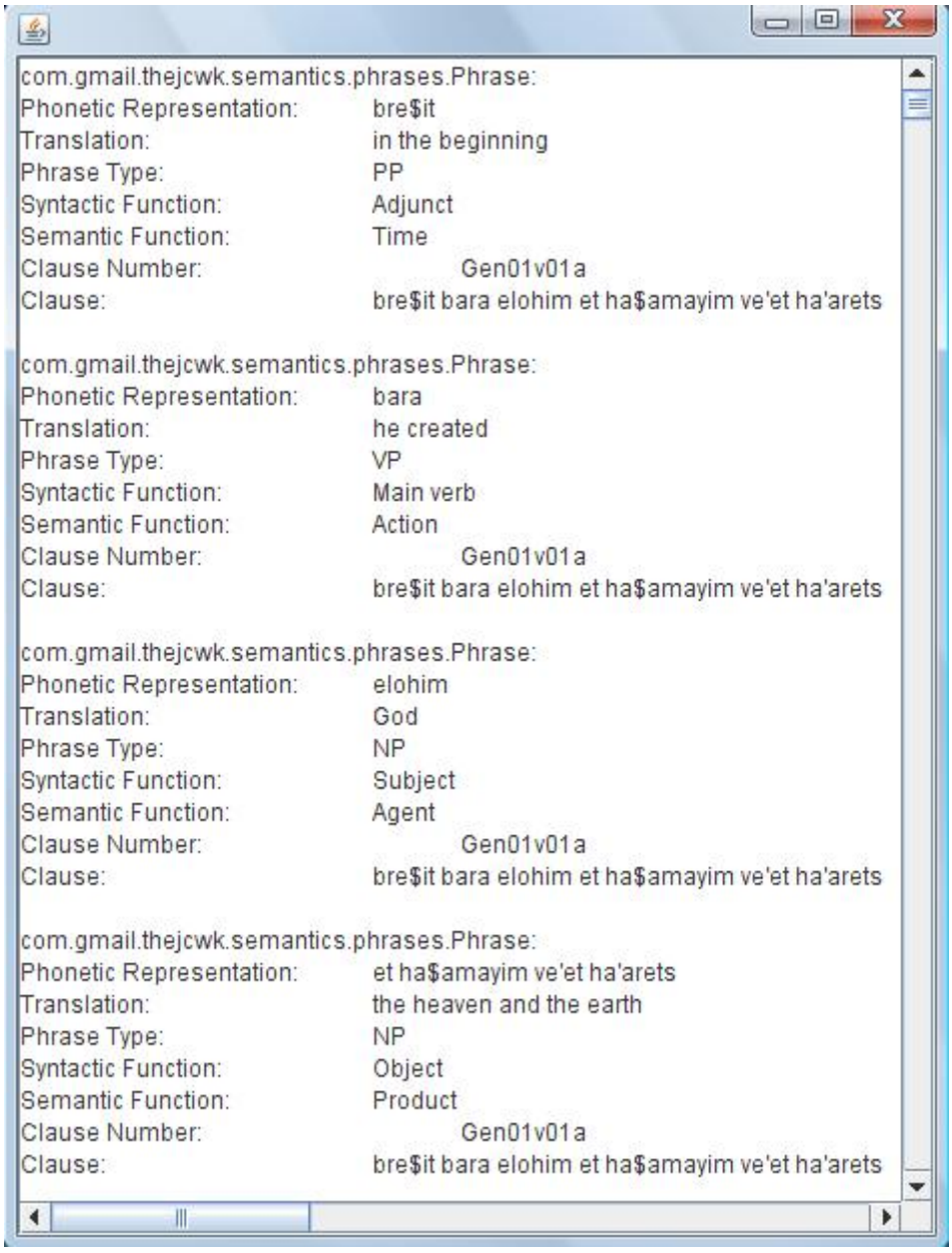

Figure 7.5. A textual representation of the phrases in the database, viewable in the visualisation program "semantics.bat".

Another, more important, option in the "View" menu is the filter management function. When the researcher clicks on "Manage Filters", a new window opens allowing the definition and fine-tuning of filters (see Figure 7.6).

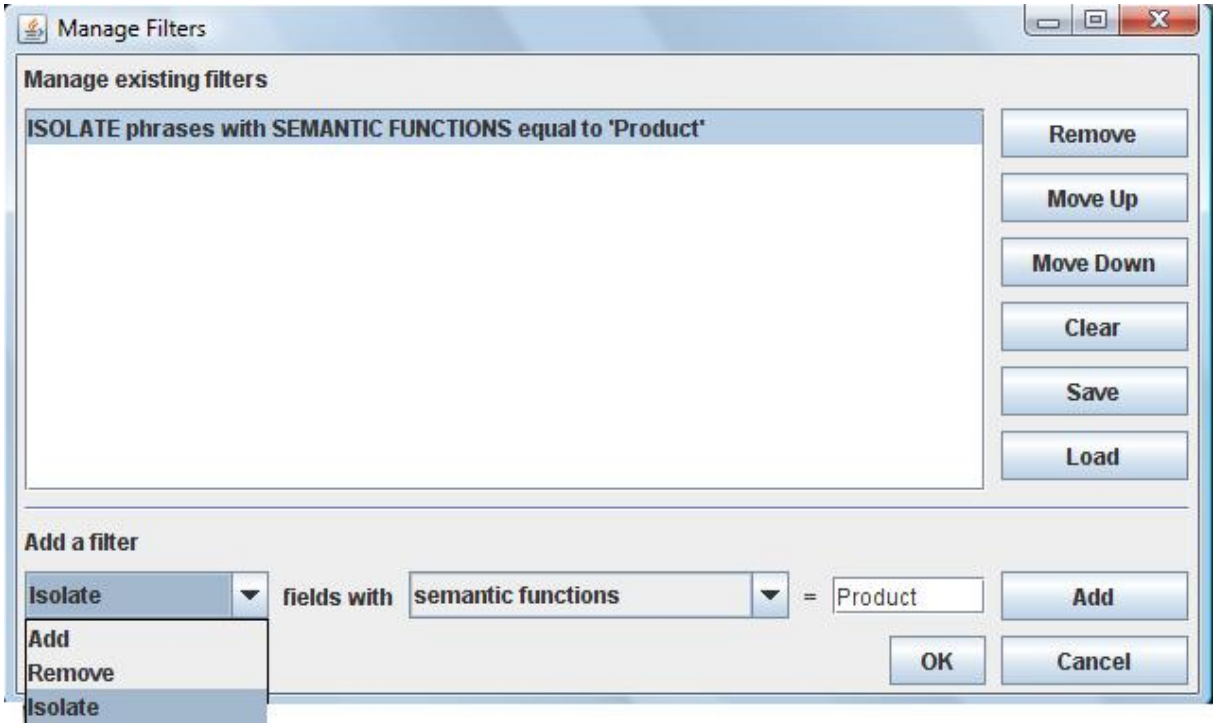

Figure 7.6. Interface used to define and fine-tune filters in the visualisation tool.

The researcher may, for example, isolate phrases with the semantic function of product by selecting the relevant options on the drop-down lists and entering the name of the required function in a textbox (located towards the bottom of the window). The filter is inserted in the window by clicking the "Add" button. The "OK" button will use the defined filter(s) to create a topic map. The results, produced by applying the current filter, are shown in Figure 7.7. It shows that, in Genesis 1:1-2:3, the semantic function of product is realised by the syntactic functions of object and complement. When the user hovers with the mouse over one of the phrases, more clause detail is shown in a pop-up window.

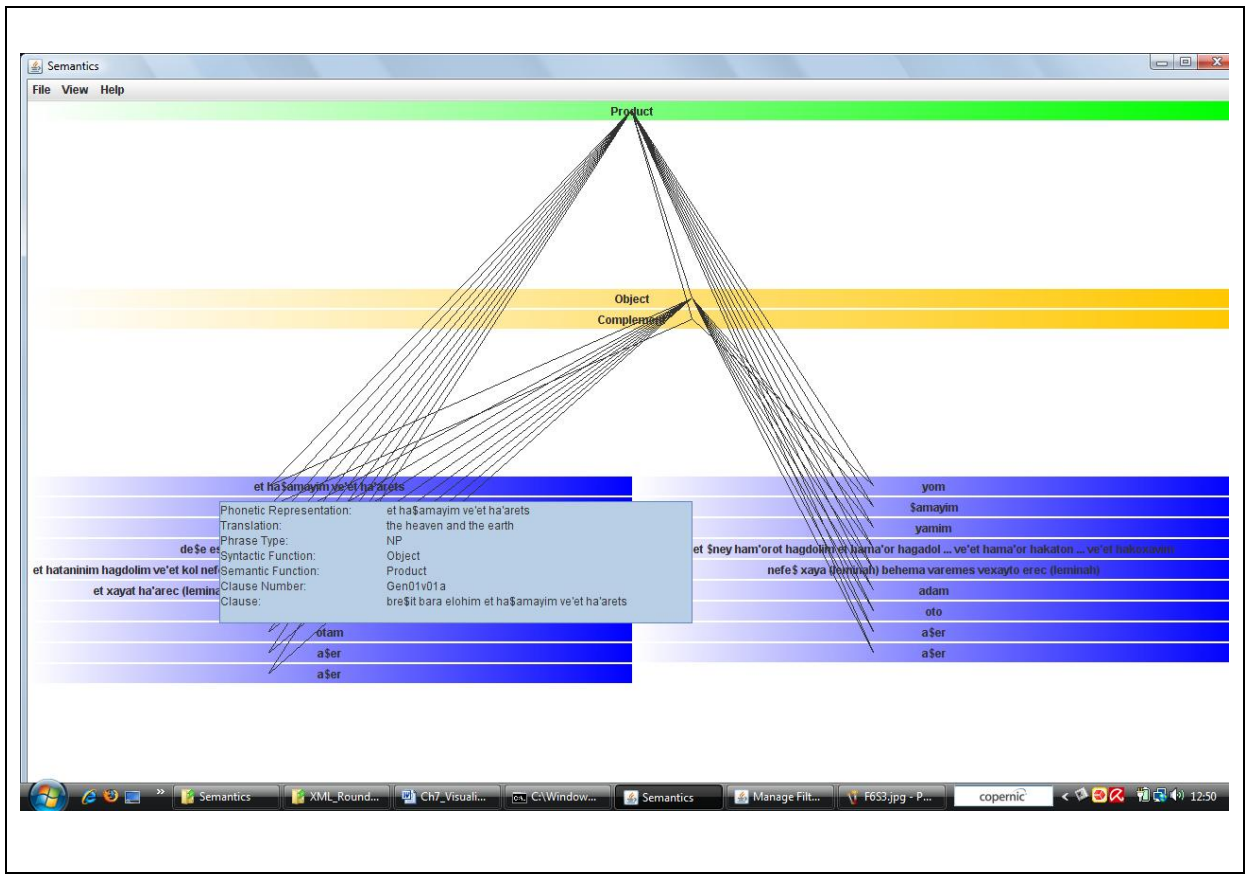

Figure 7.7. A screen shot of a visualisation of the network linking the semantic function of product to the syntactic functions of either complement or object, as found in various clauses in the dataset.

Underlying this visual representation is the slicing off of the phonetic, syntactic and semantic levels in the clause cube (see Chapter 3). To fine-tune the results, the researcher may also add or remove more filters using parameters on all three these levels. If one would like to add information on the display regarding the direct object, the following filter may be added: "ADD phrases with SYNTACTIC FUNCTIONS equal to 'Object'". The updated graphical display is shown in Figure 7.8.

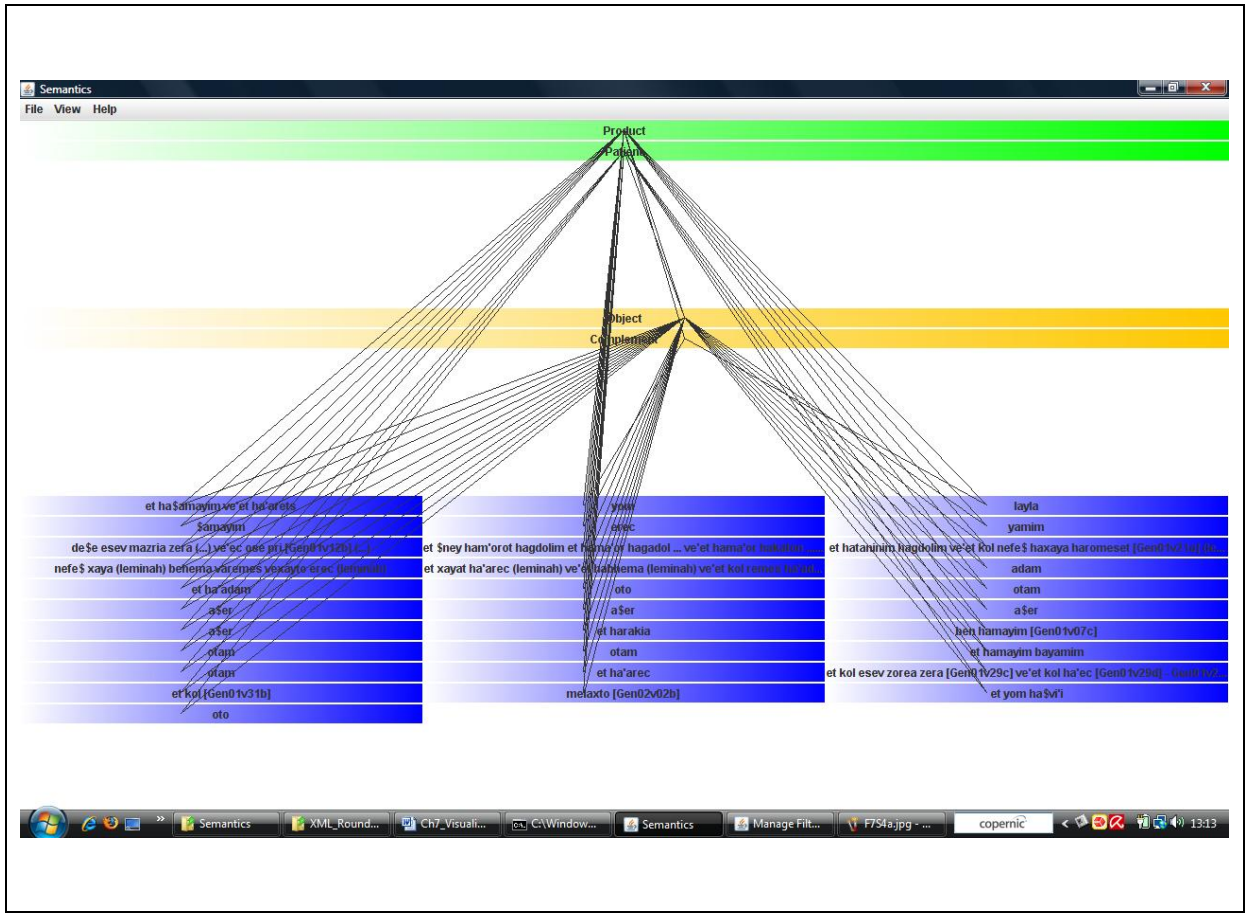

Figure 7.8. Updated graph showing the network linking the semantic functions of product and patient to the syntactic functions of complement and object, as found in various clauses in the dataset.

Since the order in which filters are applied may have an effect on the eventual output, the user is also allowed to move them up or down. A selected filter may be deleted, and even the whole filter window may be cleared to make a fresh start. If the user wants to save a filter or group of filters for later re-use, these may be saved and reloaded later (see Figure 7.6).

The help button on the default screen is currently only used to display information about the open source licence and the author of the program, but could be extended to contain user directions for researchers.

Although only one example of a visualisation tool has been implemented, it sufficiently illustrated graphical visualisation as a powerful, experimental way to search for patterns in a linguistic dataset. It directs the text-mining procedure by implementing what-if scenarios (cf. Thuraisingham, 2002: 87-89, 279).[139] Spot-checks were done and confirmed that the results of this tool match those in 6.5 (see Figure 6.25) when the same filtering parameters are used, but the visualisation tool is much more flexible. While the text-based results are limited to the testing of pre-programmed combinations, the visualisation tool allows the researcher a vast number of possibilities to add more data or to filter out irrelevant data on the display. It also provides a direct drill-down facility into the phrases and clauses matching the required conditions. The number of links also gives an intuitive indication of the frequency of this mapping. A low frequency may indicate an interesting combination or, unfortunately, a tagging error. This approach provides a more creative and interactive way of approaching the data-mining task, an endeavour which should facilitate a "careful and responsible development of the imagination" (Cilliers, 2005: 264).

7.9 Conclusion

Markus (2000: 10-11) indicated that information-system technologies are often used for purposes for which they were not initially designed (and vice versa, different technologies are used for the same purpose); for example, data warehousing was originally intended to be an enabler of data mining, but has also been applied for plain online analytical processing (without the aim to find unknown patterns or trends), for data integration purposes (as an alternative to enterprise software) and for the development of new data products. XML was created to capture metadata and formatting properties of texts. Visualisation is often used by information

[139] Mr. O.C. Kruger, a student in the B.Com. Hons. (Informatics) class of 2007 wrote a research essay that illustrates another, basic, visualisation tool regarding semantic role frames. The paper is titled "Graphical visualisation of a network of semantic frameworks in the Hebrew text of Gen. 1". The empirical part is a working program that was built in Visual Basic 2005 using an object-oriented approach. However, the functionality of the program is too limited to be used here as another enlightening example.

organisations to facilitate information retrieval.[140] However, these technologies have been used for a plethora of other uses as well. In this chapter, projects have been suggested 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. Guidelines were also given for the use of this databank, combined with visualisation techniques, to search for unknown patterns in order to create new knowledge.

The ideas discussed in this chapter and the few suggestions of visualisation implementations were submitted to make a small contribution in the search for humanistic ways of digitally exploring texts, as formulated inimitably by Sinclair (2003: 183): "I navigate through a text with the same blend of fascination, anxiety, and excitement as I explore the streets of an unfamiliar city: I do not hesitate to venture down mysterious pathways and streets, even though they may lead to a dead end. Various things along my journey may prompt me to change directions, and although I often do not know where I am going, I know that I am somehow accumulating a broader representation of the terrain. If I were given a detailed map and path to follow, I would be robbed of the enjoyment of exploration and serendipitous discovery. If I were given a list of the monuments and features of the city, I would still only have limited understanding of it. Similarly, lists of words and other components of text can be very useful and informative, but to truly experience the text I need other means of exploring it."

[140] For a list of available visualisation journals, conferences, websites and software products, see Eden (2005: 60).