TIA-2003 Conference, Strasbourg, March 31 and April 1, 2003

A Model for Dynamic Term Presentation

Oliver Streiter, Leonhard Voltmer

European Academy of Bolzano, viale Druso 1, 39 100 Bolzano, Italy
O.Streiter@eurac.edu L.Voltmer@eurac.edu

Abstract

The paper presents a model for dynamic term presentation. We break up static terminological entries into a network of elementary units. For presentation these units are assembled according to user requirements by a grammar model inspired by models of natural language generation. We show feasibility and benefits of the approach.

1. Introduction: Term Presentation as Research Topic

Most research in computational terminology has focused on term creation, storage and maintenance. The presentation of terminological data has attracted less attention, although it influences not only issues as user accessibility and user adaptation, but also term creation and term storage. If term presentation is handled intelligently and flexibly, many arbitrary decisions in term creation and storage can be postponed until user-, history- and media-related parameters allow for a well-grounded choice.

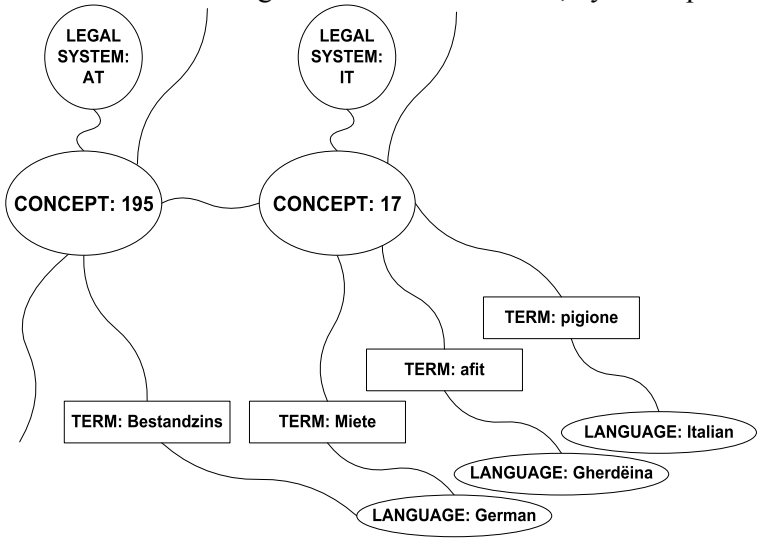
2. The Data model: Cyclic Graphs for Terminological Knowledge

The semi-structured data of terminological databases are often organized in tree-like data models. The trees or subtrees of these models are disjoint and represent autonomous knowledge units (e.g. terminological entries). The nodes in a subtree are fixed, and fixed are their hierarchical relations. Term presentation out of these models reflects the predetermined packaging and structuring. The disadvantages are obvious: First the terminographer has to make arbitrary decisions on how to package and structure information, later package and structure cannot be adapted to medium (screen, paper), user, query or other parameters. E.g. if *CONTEXT*-nodes are subordinated to *TERM*-nodes, *CONTEXT* can be accessed through *TERM*, but it is not possible to inverse this data view and to access *TERM*-nodes through *CONTEXT*. In lack of expressive power, tree-like models often resort either to redundant reduplication of data (problems of data consistency) or their implicit storage (e.g. *LANGUAGE* of a *CONTEXT* is identifiable only implicitly through *LANGUAGE* of the related *TERM*). When unrelated data shall be presented on one screen, the terminographer may be forced to fake a non existing relation for the tree-like data model.

In view of the limitations of tree-like data models in contrast to the more expressive cyclic data models, EURAC, which elaborates descriptive and normative terminological data for 5 languages in 4 countries, is using a cyclic graph-based data model, implemented as SQL relational database. Terminological data are organized in a network of nodes and edges. Nodes have a **type** (data categories like *TERM, CONTEXT*) and a **content**. In our example the con-

tent of a *TERM*-node is *pigione*. Nodes are connected by labeled edges, which express their relation (not reproduced in Figure 1).

Figure 1: A network of terminological data allows flexible, dynamic presentation



The advantages are: Data are non-redundant and unambiguously stored, partial knowledge may be stored. The information is neither pre-packaged nor pre-structured. Term presentation does not encumber the data model. Nevertheless, tree-like presentation formats can be derived from it. Terminological and lexicographical data can be jointly stored and separated through a specific data view (c.f. Sager 1990, Melby & Wright 1999).

The term presentation from a cyclic graph is however complex. In order to reduce complexity and to integrate insights from research in text generation and text linguistics (e.g. Mel'čuk 2001), we propose a four-grammar model. Every grammar handles a feasible subtask for which standard XML-based solutions are at hand.

3. The 4-Grammar-Model: A Linguistically Inspired Approach

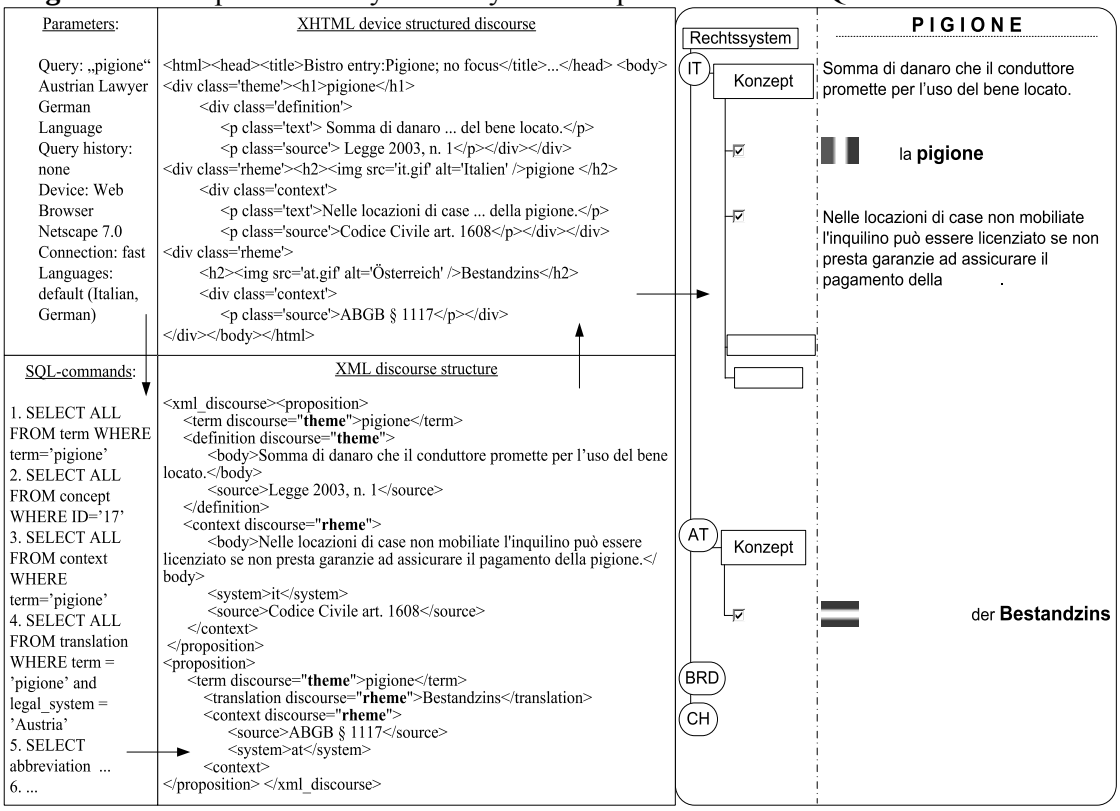
We generate views on the terminological knowledge by 4 pipelined grammar modules which react on user and setting specific parameters. The first grammar module, the **text grammar,** converts the cyclic graph into a tree structure. The **theme** of a query (*TERM:pigione*), defines the starting-point for extraction. All other nodes belong to the **rheme**. The **focus** of a query (e.g. *LEGAL SYSTEM*) determines the structure of the rheme and separates relevant from irrelevant relations. Focus-information is extracted next after the theme. Following the edges of the focus, the closest related nodes are extracted. The properties of the output tree are: 1) The root node is the **theme**. 2) The second node is, if available, the **focus** and all other extracted nodes are organized in its subtree. 3) A change of focus results in different tree structures. 4) Multiple instantiations of the focus result in flat, multi-branching trees. 5) A specific focus will result in a thin but deep tree.

The **sentence grammar** transforms this discourse-structured tree into a syntax-tree. This tree is the dynamic counterpart of the classic terminological entry. The sentence grammar selects elements for **structural markup**. Elements are *text, list, table, tree graphics, heading* and *note*. Multiple flat subtrees are prone to *lists*. Few flat subtrees would automatically come

in *tables*. Identical **node types** can be structured by diagrams. Only terminal nodes can become *notes*. The default presentation is *text.* Sentence-grammar has also a filter function on unnecessarily extracted information, where it does not fit into the presentation. Markup elements are nested to form a syntax tree (with certain combinatorial restrictions).

The **word grammar** contextualizes the content of a node. The node *LANGUAGE:English* can be presented as image of a flag, as ISO-abbreviation 'en', as 'English', 'Anglais' or '(in English)'. The parameters are discourse-structure (coming from text grammar), syntactical markup element (coming from sentence grammar), the device and interface language. The **articulation grammar** finally determines the layout. This module is sensitive to the medium, the device and the physical limitations of the user and selects font, color, size and positioning.

Figure 2: From parameters dynamically to an output screen with SQL and XSLT:



4. Techniques for Implementation: SQL, XML, XSLT, CSS, XSL-FO

Our data model can be implemented as **relational database** where **node types** are tables and **edges** are relations. The text grammar is an explicitly defined SQL-view (c.f. Ballew 1999) or a dynamically created stack of **SQL select**-statements otherwise. The first select-statement extracts the **theme**-node (e.g. "pigione") and adds the relation to the theme to the stack. The second extracts the **focus** (e.g. the definition related to "pigione") and adds all relations associated to the focus node to the stack of commands. No edge is followed twice. A parameter MAX_NODE determines the amount of data to be extracted. The output of this extraction is a specific view on the data, encoded in XML. XML has been chosen for the data transport between the database and the graphical rendering since data structure and graphical rendering can be expressed by languages of the XML family (c.f. Bourret 2003). The sen-

tence-grammar is implemented as **XSLT transformation** which creates, according to the device, XHTML, WAP, XSL-FO or SVG. The word grammar is handled mainly in XSLT through the selection among alternative labels provided as XML-attributes. The articulation grammar is expressed, according to the device, either as **CSS** or **XSL-FO**. CSS can be deselected if a user has specific accessibility requirements.

Figure 2 illustrates the 4 grammar modules. The upper left window contains the parameters: a) query term, b) user profile c) interface language d) query history, e) device information, f) channel specification g) center of interest. (a) defines the **theme**, (b) gives preference to conceptual information, (c) gives the interface language (d), if applicable, renders information coherent to previously presented information, e) determines the XML syntax to be selected, packaging and layout, f) determines packaging and layout and g) filters out unwanted information. The parameters trigger SQL-commands (window below) which extract: 1) the **theme** "pigione" 2) the most pertinent relation according to the parameters 3) the *CONTEXT* to the term 4) comparative information from the Austrian legal system. The query results are transformed to XML (down middle window). The structure renders a specific view on the data base. This discourse structure is adapted to the user device (top middle window). Concept trees, language flags and HTML text are produced and arranged for web-presentation. The right side shows one possible output.

5. Summary and Conclusions

Starting from the need to improve the presentation of terminological knowledge we compared data models for terminological data. We could show that organizing data in cyclic graphs not only solves most problems related to term presentation, but also problems of term storage. An important advantage for term storage is that terminological data and lexicographical data can be stored in one knowledge base. The relational data model is sufficiently powerful and already implemented in mature and free databases (c.f. Holmes-Higgin & Khurdshid 1996). For data transport and transformation we use XML. XML absorbs the database output and escorts the data through XML transformations until its final rendering in pixels or ink points. The data transformation is piloted by four grammar modules which epitomize linguistic models of knowledge communication. Future research will optimize the grammar modules and their interaction.

References

BALLEW, R., DUNCAN T. AND BLASINGAME M. (1999), *Relational Data Structures for Implementing Thesauri*, http://www.mip.berkeley.edu/mip/related/thesaurus/thesaurus.pdf

BOURRET, R. (2003), *XML and Databases*, http://www.rpbourret.com/xml/XMLAndDatabases.htm

HOLMES-HIGGIN, P. and KHURSHID A. (1996), *Is your Terminology in Safe Hands? Data Analysis, Data Modelling and Term Banks, Terminology and Knowledge Engineering*. August, Vienna: Austria.

MELBY, A.K., WRIGHT S.A. (1999), *Leveraging terminological data for use in conjunction with lexicographical resources.* Terminology and Knowledge Engineering. June-August, Innsbruck, Austria.

MEL'ČUK, I. (2001), *Communicative Organization in Natural Language: The Semantic-Communicative Structure of Sentences*: John Benjamins Publishing Company.

SAGER, J. C. (1990), *A practical Course in Terminology Processing*. Amsterdam: John Benjamins.