

An example of using electronic dictionary tools aimed to build language teaching exercises¹

Dainis Zeps

Institute of Mathematics and Computer Science,
University of Latvia
Riga, Latvia
dainize@mii.lu.lv, <http://lingua.id.lv>

Extended Abstract

Most of all general aims of this conference are outstanding and urgent in field of education of languages, and ancient languages including. Author's intention is to add his worry and his activity in general tend to cure the situation. On his own example author wants to discuss more general questions too. Authors [Latin Dictionary Tools Page](#) [3] comprises in itself database equipped with functions aimed to investigate possibilities to use these functions in building exercises in learning Latin.

Let us equip a simple Latin sentence with links to database: [catus captat musculum](#) linking forms with inflection generator. The same sentence may be linked with word meaning and form explaining function: [catus captat musculum](#). Alas, html does not support multifunctional hyperlinks... Question 1: Name feminine nouns of 4. declination: Look in [simple](#) dictionary or in a [larger](#) (in combo-boxes)! Question 2: Change form [musculum](#) in plural: [Look!](#) Author uses these and similar samples of exercises in teaching Latin in his e-courses in university.

Some very general suggestions. Language html allows only hyperlink with one function link. To build *multilinks* similar to pop-down menus or more general, where its functionality could be user defined, one must implement in a program unit written in one of general programming languages and then use it as a functionality built in this program. Of course such functionality would be desired for some specimen of html in future. But let this serve us for an example for many other such possibilities that are not possible in html or similar language environments. Speaking in terms of computer science the functionality of multilink as described above can be defined in terms of mathematic machines and there would exist a mathematical machine called *multilink*. Contemporary programs are built in terms of mathematical machines, but it applies only what concerns what proceeds within program itself, but extremely weakly [only what requires operational system in general] what concerns its outside. If we had both directions then we could build sequences of mathematical machines to build what could be named user built programs. The last is very badly developed and supported by developers and maybe because developers are interested in their products as some end-unities but not as bricks in some more general programs where in its production may take part many different producers or rather all producers. From part of users, **there should work a principle that good program is only such that may be used as corporative link in some more general programs, i.e. this program is a mathematical machine outside too.**

Tutorial of languages in former classical form should be turned into program environment where grammar rules and categories have mathematical machines in correspondence, where the rules that incorporate these notions in language have corresponding rules that incorporate corresponding machines in the general machine. Unless this we have it is hardly to hope to cope with languages.

Language environments are most necessary for human use but most of all in a state of neglect. It seems that a general rule works: learning of languages is mostly hard paradigm for human mind, and, consequently, non recognition of our weakness in understanding languages places us in where we are. Maybe it must be accepted this as a general paradigm and await some result only when adequate attention is turned to it and, reversely, not to expect any result unless we recognize weak application of effort by us in that direction.

Bibliographical References

- Acho A., Hopcroft J., Ullman J. 1974. *The Design and Analysis of Computer Algorithms*. Addison Wesley.
The Oxford Handbook of Computational Linguistics, R.Mitkov, ed., University Press, Oxford, 2003.
D. Zeps 2004. [Latin Dictionary Tools in Internet](#), in [Human Language Technologies](#), Riga, Latvia, 2004, 179-182. [<http://susurs.mii.lu.lv/dainize/lingua.htm>]

¹ In order to follow links use electronic form of the article.