

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. 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.

Teachers of languages are these which need tools to build exercises and learning tools; but problem is that teachers do not know what they could have if mathematicians could guess their needs. What we have reflects that what dialog between both sides was possible to give up to now. But we see more problem in the fact that both sides poorly realize what other side needs; the situation making worse fact that both have a sense about what other side needs mostly being delusive. Industrial programs show how poor the situation is.

We think in the direction to try to find new paradigmata to invent dynamic tools that would replace the text book of teacher changing it in a completely dynamic tool. The language and its grammar should be that general machine which have several sets of functions where these of teacher and student are clearly discernable: language machine [eventually more or less common for all languages] would have some simple kernel but extended to its natural size by all its irregularities but not losing its generality. The learning process can be imagined as an interface between text and whom that should the text understand: the unknown text is that should be unveiled to the student through tutorial environment where the language machine is 'in a fly', i.e. in all the functionality. It would be a progress to encapsulate all learning process in a similar scheme.

For now teachers have only mostly general tools e.g. MS office tools e. c. to build their exercises but almost nothing is specifically developed for languages: *all must be programmed*. We have an acute need in more survivals such as [Perseus project](#), but even more, tools to build learning tools and some people who could help in it. Every project facilitating it should be appreciated. Author has tried to help himself in his teaching praxis by building such tools as [Reading exercises in Latin](#) to read Faedrus in Latin and [Latin Dictionary Tools page](#). Last tool was invented with idea to search after new ideas in understanding old language terms.

Author is building exercises using this tool in teaching Latin in his e-courses in university: 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!](#).

A very general suggestion. 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.**

Bibliographical References

1. Aho A., Hopcroft J., Ullman J. 1974. *The Design and Analysis of Computer Algorithms*. Addison Wesley.
2. *The Oxford Handbook of Computational Linguistics*, R.Mitkov, ed., University Press, Oxford, 2003.
3. D. Zeps 2004. [Latin Dictionary Tools in Internet](#), in [Human Language Technologies](#), Riga, Latvia, 2004, 179-182.

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