

# JAPASSIST: AN ASSISTANT FOR STUDYING JAPANESE WHILE SURFING THE WEB

Shiori Daigo<sup>1</sup>, Rosario Giunta<sup>2</sup>, Giuseppe Pappalardo<sup>2</sup>, Emiliano Tramontana<sup>2</sup>

<sup>1</sup>Facolta' di Lingue e Letterature Straniere

<sup>2</sup>Dipartimento di Matematica e Informatica

Università di Catania

Keywords: Web personalisation, user assistance, e-learning.

Abstract: The web offers an increasing amount of pages written in several languages. Given the variety and the quality of some pages, these can be helpful as an additional language learning support. Indeed, when studying a foreign language, reading newspapers, magazines, etc. is one of the preferred ways for learners to improve their abilities. However, without an adequate support they would be forced to make extensive use of the dictionary.

This paper proposes Japassist, a browsing assistant that supplies users with a customisable amount of suggestions, looked up into dictionaries, useful to read and study the Japanese language while surfing the web. Japassist aims at supporting not only users who need a beginner-level understanding of Japanese documents, but also those having some comprehension of the Japanese language. Actually, given the Japanese language intrinsic complexity, some help to read or appreciate the meaning of uncommon kanji symbols may be useful even for native speakers.

We have employed the *aspect-orientation* technology to customise a bare web browser with the ability to assist users reading and studying Japanese. This design approach allows the web browser classes to be unaware of assistance classes, thus promoting low coupling between different concerns.

## 1 INTRODUCTION

Given the large amount of web pages written in several different languages, it is generally useful to provide web users with suggestions about the meaning of foreign text in visited pages. As it happens when studying a foreign language, reading the text available from web sites offering newspapers, magazines, etc. greatly helps learners to improve their abilities. Moreover, the electronic version of a page is ideally suited to be customised, with information looked up into dictionaries, so that learners can enjoy a promptly available support.

This work proposes Japassist, a browsing assistant that supplies users with a customisable amount of suggestions useful to read, translate and study Japanese pages while surfing the web. Japassist runs on the user host, which makes it is easy for the user to activate, deactivate, or choose the desired amount of, suggestions.

Support for understanding a foreign language web page is only limited currently to single word translation (Google, 2005b) and (often misleading) whole

text translation (Google, 2005a; Altavista, 2005).

Several features make Japassist assistance different from other supports: (i) words can be separately listed and explained; (ii) words and characters can be highlighted to ease user reading and understanding; (iii) text can be supplemented with reading assistance for kanji<sup>1</sup>, hiragana<sup>2</sup> and katakana<sup>3</sup>; (vi) Japanese words, as well as their composing characters, can be enriched with more than one English translation and suggestions on the context where they should be used; (v) grammar particles can be highlighted, thus giving hints on grammar rules regulating text; finally, (vi) at the architectural level, the use of *aspect-oriented programming (AOP)* (Kiczales et al., 1997; PARC, 2005) affords the ability to develop a personalised web browser, without affecting the original browser

<sup>1</sup>System of Japanese writing using Chinese characters. A kanji can be a word on its own or a part of a word.

<sup>2</sup>System of syllabic writing for Japanese, especially used for function words and inflections.

<sup>3</sup>System of syllabic writing for Japanese, especially used for words of foreign origin.

source code with changes.

As far as the software architecture is concerned, a simple Java browser has been enhanced by connecting it with Japassist assistance functionalities. Connection is obtained by means of AOP, thus maintaining the two main concerns we have focused on, i.e. browsing and assistance, clearly separated.

Thanks to the noted assistance functionalities, Japassist is useful not only as an automatic tool for translation, but also as an assistant for users wishing to learn, or improve their knowledge of, Japanese, while surfing. Actually, given the Japanese language intrinsic complexity, users having a certain comprehension still need help for understanding many other kanji (among the around 50 thousand existing kanji (Kubota, 1989)). Moreover, some support to read or appreciate the meaning of uncommon kanji and words is useful to native speakers, or to advanced level learners.

The sequel of the paper is as follows. Next section presents the detail of assistance functionalities. Section 3 describes the software architecture of Japassist and its connection to a web browser. Section 4 examines the related work. Finally, section 5 draws some conclusions.

## 2 ASSISTANCE FUNCTIONALITIES

Japassist assistance functionalities can be grouped in three categories: (1) counting of kanji and words occurrences, (2) kanji and words highlighting, and (3) suggestions about reading and English meaning. These are described in the following three subsections respectively.

### 2.1 Counting Kanji and Word Occurrences

Japassist holds and updates the count of occurrences of kanji and words as two ranked lists, i.e. *kanji-count* and *wordcount*, to reflect the web pages visited by the user. The *kanji-count* and *wordcount* lists count how many times each kanji and word, respectively, has been found while surfing the web. In a customised way, these two lists suggest users the most useful kanji and words. This is similar to what the Halpern dictionary (Halpern, 2003) provides, i.e. providing ranked lists of words and kanji on the basis of their use in daily newspapers.

Figure 1 shows the dialogue window that, upon user request, reports the *wordcount* and *kanji-count* lists, in the left and centre panels, respectively.

On this dialogue window, the user can select a word or kanji, in the left and center panel respectively, to obtain in return the following information:

1. in the bottom panel of Figure 1, the reading and meaning in English, gathered by looking up the electronic *EDICT*<sup>4</sup> (Breen, 2005a) and *KANJIDIC*<sup>5</sup> (Breen, 2005b) dictionaries for words and kanji respectively;
2. in the right panel of Figure 1, the list of existing words that use the selected kanji, produced by scanning the *EDICT* dictionary for a word including the kanji.

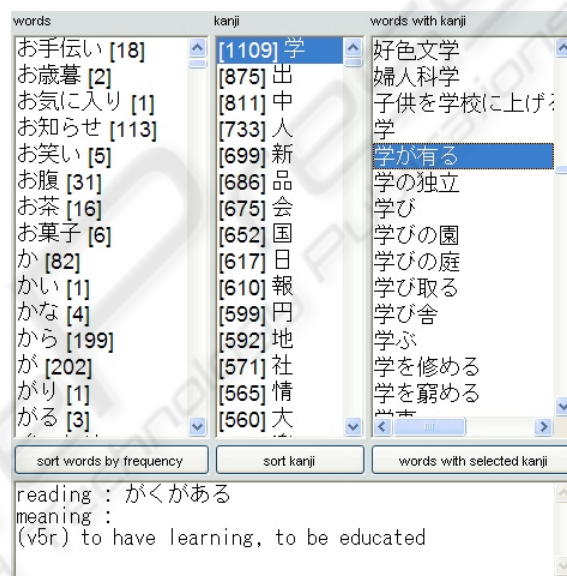


Figure 1: Counting kanji and word occurrences.

### 2.2 Kanji and Word Highlighting

The aim of characters and word highlighting is to make it easier for learners to remember the most important kanji and to suggest the usage of grammar particles and character sets. Four different highlighting modes are at user disposal: (i) *Jouyou* and *Jinmei* kanji, (ii) customised list of kanji or words, (iii) grammar particles, and (iv) character sets colouring.

<sup>4</sup>EDICT consists of a text file in which a row is an entry of the dictionary. Each entry contains a word, its reading and meaning. Most of the words have a single reading and meaning. However, words with multiple readings and/or meanings are dealt with by separate entries.

<sup>5</sup>Similarly to EDICT, in the KANJIDIC file each row/entry contains a kanji, its readings, meanings and some additional data such as its frequency (Girardi, 2005). Each kanji is found in one row only.



Figure 2: Kanji highlighting, furigana reading and translation tooltip.

In March 1981, the council of Japanese language submitted the list of 1945 kanji, called Jouyou kanji, that are most used in daily life, press, broadcasting, law and official documents; and the list of 166 Jinmei kanji that are used for names, which became 284 in 1992 (Shinmura, 1994).

In highlighting mode (i), when a web page is requested by the browser, Japassist finds all Jouyou and Jinmei kanji in it and sets their background colour to green. The background colour is changed by adding the span tag (World Wide Web consortium (W3C), 2005a) around the kanji into the HTML source of the visited page.

The background colour varies from light green to dark green according to the kanji classification, i.e. the kanji that are usually learnt in the lower classes of elementary school have a darker background than those in higher classes (information on which classes a kanji is taught in is available in KANJIDIC). Figure 2 shows an example of the green background of kanji and words on the Yahoo! Japan home page.

In highlighting mode (ii), Japassist marks as light grey the background of a user defined list of kanji or words. This list is initially set by Japassist, with the most frequent kanji found in the visited web pages (see *kanjicount* in section 2.1), and can be modified

by the user through a provided front-end.

As for highlighting mode (iii), yellow is used to mark the background of hiragana grammar particles. These particles are used for distinguishing the role (subject, object, etc.) of nearby words (see e.g. the yellow background colour of hiragana in Figure 2).<sup>6</sup>

Finally, in order to help users determine to which writing system, i.e. kanji, hiragana or katakana, a character belongs to, Japassist can be set to mark the text using an associated colour for the foreground. E.g. we have chosen black for kanji, blue for hiragana and dark grey for katakana. Figure 3 depicts a rendered web page fragment whose foreground colour reflects whether text is in the hiragana or kanji script, while background colour highlights grammar particles and Jouyou and Jinmei kanji.

Beginner-level learners can take advantage of text colouring, since a typical difficulty they face is to distinguish between character sets.

<sup>6</sup>The current version does not use a grammar analyser, therefore some hiragana, used as a phonetic character, could be mistakenly recognised as a particle.

## つまどひ

世界の男、貴なるも賤しきも、いかで、この赫映姫を得てしがな見てしがな、と音に聞きめてて惑ふ。その邊の垣にも家の外にも居る人だに、容易く見るましきものを、夜は安き寝もねず、闇の夜に出でても穴を抉り、此處彼處より覗き垣間見惑ひあへり。さる時よりなむ、よばひとはいひける。

人の物ともせぬ處に惑ひ歩けども、何の驗あるべくも見えず。家の人どもに物をだに言はむとて、言ひ懸くれども、事ともせず。邊を離れぬ公達、夜を明し日を暮す人多かり。疎かなる人は、益なき歩行はよしなかりけりて、來ずなりにけり。その中に猶言ひけるは、色好といはるゝ限り五人、思ひ止む時なく夜書來けり。その名、一人は石作皇子、一人は車持皇子、一人は石大臣阿倍御主人、一人は大納言大伴御行、一人は中納言石上麻呂、只この人々なりけり。世の中に多かる人をだに、少しも容貌よしと聞きては、見まほしうする人々なりければ、赫映姫を見まほしうして、物も食はず思ひつゝ、かの家に行きて、佇み歩きけれども、詮あるべくもあらず。文を書きてやれども、返事もせず、侘歌など書きてやれども、返しもせず、詮なしと思へども、十一月十二月の降り氷り、六月の照り曇くにも障らず來けり。この人々、或時は竹取を呼び出でて、「娘を我に賜へ」と伏し拜み、手を擦り宣へど、「己がなさぬ子なれば、心にも従はずなむある」といひて、月日を過す。

Figure 3: Hiragana and kanji highlighting.

### 2.3 Suggestions for Reading and English Meaning

According to user preferences, Japassist can decorate the text of visited web pages with help information, to supply, both for words and kanji, some aid in terms of reading, transliteration, English translation or English semantic explanation. These are orderly described in the following items.

- Reading assistance is available for kanji and can be written in hiragana and katakana. This can be useful for learners that can read hiragana and katakana, but have a scarce familiarity with the harder kanji script. Kanji can be read in three different ways: *on*, *kun* and *nanori*. Our support can be set to provide the three different readings.

Figure 2 shows several examples of reading suggestions for kanji provided by *furigana*, i.e. (smaller) text inserted on top of a word or kanji.

- Transliteration aims at providing the phonetic equivalent of hiragana, katakana and kanji into *romaji*, i.e. roman writing. When the transliteration suggestion has been selected, the Japanese text of the visited web page is substituted with the transliterated version, as shown e.g. in figure 4. In this transformation, the shape and structure of the document remain almost unaltered.

Transliteration in romaji is useful for beginner-level learners that are not acquainted yet with the hiragana and katakana writing systems.

- English translation assistance is available for words written in hiragana, katakana and kanji. It provides users with the corresponding English word or the explanation of the meaning in English, according to EDICT and KANJIDIC. The user can set whether s/he wants to obtain the main or all the meanings corresponding to a kanji or a word.

Since each kanji can have several meanings, the appropriate one is obtained by relating a kanji with contiguous ones, e.g. our support provides the translation corresponding to the longest set of kanji found into KANJIDIC.



The screenshot shows a web page with a search bar and a list of search results. The search results are as follows:

kau	SHOPPINGU	OOKUSHON	KOMIKKU
	CHIKETTO	ryokou	shucchou
	takuai	kessai	KONTENTSU SUTOA
shiru	NYULUSU	tenki	SUPOOTSU
			FAINANSU

Figure 4: Transliteration of a web page in romaji.

Once a combination of assistance suggestions are activated the user can surf through several web pages and automatically be supported by the selected assistance. The various assistance suggestions are at the user's disposal and, upon selection, they can be pro-

vided in three assistance modes: *tooltip*, *furigana* and *append* mode.

The *tooltip* assistance mode provides the user with a small yellow window containing additional information when the mouse is pointed over a kanji or a word. This assistance mode allows Japassist to supply suggestions unobtrusively, i.e. the web page remains unaltered, and is meant for advanced level learners that need not have a large amount of suggestions. Figure 2 shows a web page including a translation tooltip.

The *furigana* assistance mode uses *ruby* HTML tags (World Wide Web consortium (W3C), 2005b) to insert on top of a word or a kanji the additional supporting text (this additional text is said furigana). This assistance mode, unlike the previous one, does not conceal the additional text, so that the web page format becomes slightly different than the original page. This assistance mode is more suitable for medium-level learners. Figure 2 shows a web page including several fragments of furigana text providing reading suggestions.

The *append* assistance mode provides the additional text by mixing it with the original text. Suggestions are added just after the word or kanji they refer to, and both the word or kanji and the suggestions are enclosed within square brackets. Although *append* mode largely modifies the original page layout, it can be particularly useful for beginner-level learners that need a large amount of suggestions.

### 3 ASPECT ORIENTED SOLUTION

The two main concerns we deal with are *browsing* and *assistance*. By means of the proposed aspect-based software architecture these concerns are clearly independent, i.e. no classes implementing the first concern should be aware of classes implementing the second concern and vice versa. *Browsing* and *assistance* concerns are connected to each other by means of the *AddOn aspect*. This aspect provides the desired ability to develop a personalised web browser without inserting any changes into the source code.

Japassist assistance functionalities have been implemented as a namesake Java package, whose classes receive a HTML page and enrich it with the additional data corresponding to the desired suggestions. The JBrowser package implements browsing functionalities.

JBrowser behaviour is customised with the results Japassist provides by means of the *AddOn aspect*. According to the aspect-oriented approach, *AddOn* specifies the JBrowser operations, e.g. method invocations, that need to be connected with the methods of

Japassist. Figure 5 shows an overview of the architecture and highlights the classes of the two packages that are connected to each other by the *AddOn aspect*.

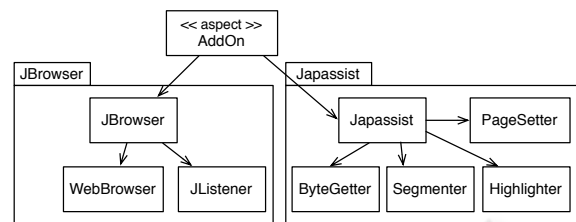


Figure 5: Aspect-oriented architecture for JBrowser and Japassist.

#### 3.1 Application Concern JBrowser

JBrowser is a simple Java web browser, consisting of classes that handle, for a given web address, page downloading, rendering, etc.

Rendering is based on an existing component, such as Microsoft Internet Explorer or Mozilla for Windows, and Mozilla for Linux. JBrowser connects with the chosen non-Java rendering component by means of the Java Desktop Integration Components (JDIC) libraries (Java Community, 2005). Within JBrowser, the class *WebBrowser* of package *org.jdesktop.jdic.browser* provides a blank area into which web pages can be displayed by invoking the *setURL(aUrl)* method. In turn, this class invokes native methods (through JNI) of the available browser.

In addition, the said JDIC package provides interface *WebBrowserListener* that allows an instance of the class implementing this interface to be notified when some events occur on the *WebBrowser* instance it is registered to. In this context, class *JListener* is an implementation of *WebBrowserListener* and its methods e.g. *downloadStarted()*, *downloadCompleted()*, etc. are invoked when the corresponding events occur.

#### 3.2 Assistance Concern Japassist

The assistance functionalities of the package Japassist are provided by classes *Japassist*, *PageSetter*, *Segmenter*, *Dictionary*, *Highlighter*, *OccurDialog* and *Occurrences*.

Class *Japassist* is implemented on the basis of the *Facade* design pattern (Gamma et al., 1994) and is responsible to invoke the methods providing the user requested assistance support within the visited web page.

*PageSetter* is intended to change the original HTML page source in order to insert into existing

links their absolute path. This allows links within the local enriched copy to still work correctly.

Segmenter provides methods allowing a sequence of characters, taken from the web page, to be split into words. Since in Japanese text words are not separated by blanks, in order to detect whether a sequence of hiragana, katakana or kanji constitutes one word, we need to search the longest sequence of characters that matches a word listed in EDICT.

Dictionary holds the EDICT and KANJIDIC dictionaries, and the list of grammar particles. Its methods make it possible to look up words and kanji, and return their readings and meanings.

Highlighter enriches HTML source by adding tags allowing text to be coloured (cf. section 2.2). It also manages tooltips.

OccurDialog is responsible to draw the window showing word occurrence (cf. Figure 1) taking as input the data organised as class Occurrences. OccurDialog is instantiated when the user clicks on the corresponding button of the browser window.

### 3.3 Aspect AddOn

Aspect AddOn is responsible to connect the browser with assistance activities.

AddOn defines pointcut `init()` that is responsible to insert some buttons and checkboxes into the browser front-end when an instance of `WebBrowser` is created. These buttons allow the user to: (i) select the desired suggestions that will be inserted into pages while surfing the web; and (ii) interact with additional `Japassist` dialogue windows (e.g. cf. section 2.1).

At run-time, `Japassist` should insert suggestions into visited web pages when their download is finished. For this, `AddOn` defines pointcut `completed()`, see listing in Figure 6, that captures method `documentCompleted()`.

```
pointcut completed() :
    execution (public void
                documentCompleted(WebBrowserEvent));
```

Figure 6: A pointcut capturing the termination of pages download.

Figure 7 shows the sequence diagram for the activities that are started once a page download has completed. The defined `AddOn after Advice`, i.e. the code that is executed after the pointcut has captured control from `JBrowser`, provides means to bring control to class `Japassist` that starts a set of activities. Firstly, the bytes corresponding to the HTML page are extracted by means of a supporting class `ByteGetter`. Secondly, `PageSetter` is invoked to make existing links absolute. Then, class `Japassist` invokes

other methods to insert the assistance text into the web page. In the example of Figure 7, highlighting is obtained by invoking class `Highlighter`.

Finally, the modified page returns to class `Japassist` that updates the original page, rendered by the browser, with the one enriched with suggestions.

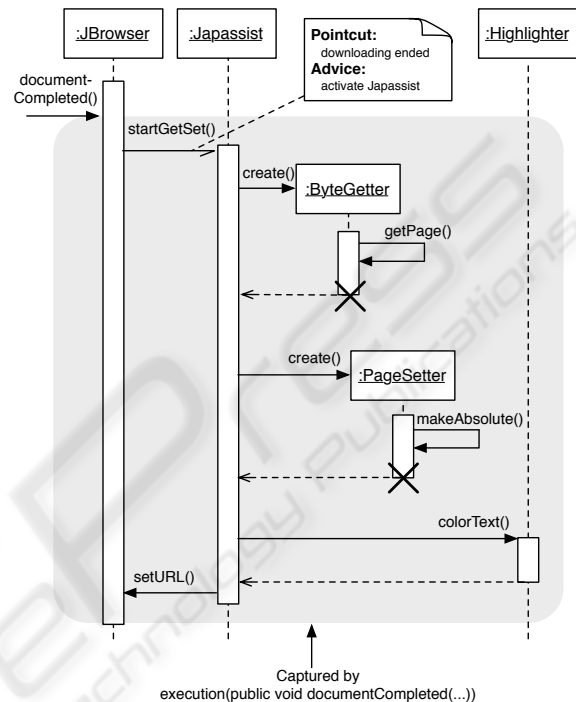


Figure 7: Sequence diagram for the assistance activities.

## 4 RELATED WORK

Google Language Tools (Google, 2005a) and Babelfish (Altavista, 2005) are some examples of services available on the web that provide text translation from Japanese to English. Once such a translation service has been requested, it can remain active to operate on the following visited web pages. These services are meant as a translation support rather than an e-learning tool like `Japassist`.

Googlebar (Google, 2005b) is a tool that provides single word translations, by means of a tooltip, from English to other languages, including Japanese. In general, this is more suitable to Japanese native speakers than to Japanese learners. Japanese learners having had an English word translated into kanji would need some further hints, such as e.g. how to read it, what are the related meanings, how to use it. Thus, this tool has a quite limited e-learning support for Japanese.

JquickTrans (Systran, 2005) is an e-learning tool that allows users to create the kanji and word lists they need to study. This tool uses EDICT and KANJIDIC dictionaries. The aim of JquickTrans is to provide an advanced electronic dictionary with useful annotation facilities, which is quite different from enriching web pages with e-learning support.

The Japanese WorkBench project (Winiwarter, 1999) is an e-learning tool for the Japanese language that, using a web-interface, provides support for reading and understanding kanji and words. A text is transformed by WorkBench so that each kanji and word is provided with a link that, upon clicking, fills a separate frame with readings and meanings.

Compared with the latter tool, Japassist affords a more advanced support, in that: (i) its assistance functionalities are more numerous and dynamically available while surfing the web; (ii) selected suggestions can be set to interfere as little as possible with the text and the user's reading activity; and (iii) additional dialogue windows are available to list useful and frequent words and kanji.

At the architectural level, the use of aspect oriented programming avoids any interference between assistance and browsing functionalities, hence any need to modify or adapt browser code.

## 5 CONCLUSIONS

The Japassist e-learning support for Japanese language has a rich set of features that can satisfy the needs of beginner and advanced level learners. These features range from counting occurrences to text colouring and highlighting, as well as explaining the meaning of kanji and words into English. All the e-learning aid is available while browsing the web on-line by enriching the visited web page.

A simple Java browser has been enhanced and connected with Japassist assistance functionalities through an appropriate *aspect*. This provides means for the two main concerns we have focused on, i.e. browsing and assistance, to be kept clearly separated.

Performance of Japassist strongly depend on the amount of required suggestions and on the length of the web page. On a 2.6GHz Pentium 4, 1GB RAM, host updating the HTML source of Yahoo! Japan home page to have the several suggestions shown in figure 2 takes around 3 seconds. However, the original page is rendered once download is completed, and suggestions are added in the background, thus the said delay often does not affect the user.

The preliminary experimental tests with Japanese language learners have shown that the provided aid is very appreciated. By using Japassist, learners find it easier to read and comprehend Japanese text both

with teachers and on their own. Besides reducing time and effort for manually looking up dictionaries, the adjustable amount of suggestions allowed learners to fine-tune the extent of additional text, letting them focus on just what they needed.

## REFERENCES

- Altavista (2005). Babel Fish Translation. WWW. <http://babelfish.altavista.com>.
- Breen, J. (2005a). EDICT Project Home Page. WWW. <http://www.csse.monash.edu.au/jwb/edict.html>.
- Breen, J. (2005b). KANJIDIC Project Home Page. WWW. <http://www.csse.monash.edu.au/jwb/kanjidic.html>.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, R. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. Reading, MA.
- Girardi, A. (2005). Word Frequencies. WWW. <http://ftp.cc.monash.edu.au/pub/nihongo/wordfreq>.
- Google (2005a). Google Language Tools. WWW. [http://www.google.com/language\\_tools](http://www.google.com/language_tools).
- Google (2005b). Google Toolbar. WWW. <http://toolbar.google.com>.
- Halpern, J. (2003). *The Kodansha Kanji Learner's Dictionary*. Kodansha International.
- Java Community (2005). Java Desktop Integration Components (JDIC). WWW. <http://javadesktop.org/articles/jdic/>.
- Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J. M., and Irwin, J. (1997). Aspect-Oriented Programming. In *Proceedings of the 11th European Conference on Object-Oriented Programming (ECOOP'97)*, volume 1241 of *Lecture Notes in Computer Science*, Berlin, Germany. Springer-Verlag.
- Kubota, Y. (1989). *Grammatica di Giapponese Moderno*. Ca Foscarina.
- PARC (2005). AspectJ Project. WWW. <http://www.parc.com/research/projects/aspectj/>.
- Shinmura (1994). *Koujien, 5th edition*. Iwanami Shoten.
- Systran (2005). JquickTrans. WWW. <http://www.coolest.com/jquicktrans/>.
- Winiwarter, W. (1999). A Language Learning Environment for Assisting Foreigners in Reading Japanese Web Pages. In *Proceedings of the 5th International Congress on Terminology and Knowledge Engineering*, Vienna.
- World Wide Web consortium (W3C) (2005a). HyperText Markup Language (HTML). WWW. <http://www.w3.org/Markup/>.
- World Wide Web consortium (W3C) (2005b). Ruby Annotation. WWW. <http://www.w3.org/TR/ruby/>.