WO 00/79427 A2

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date

(21) International Application Number:
PCT/US00/40257

(22) International Filing Date:
21 June 2000 (21.06.2000)

(25) Filing Language:
English

(26) Publication Language:
English

(30) Priority Data:
09/337,178
21 June 1999 (21.06.1999)
US

(71) Applicant: CLEVERLEARN.COM [US/US]; Suite 230,
2901 28th Street, Santa Monica, CA 90405 (US).

(72) Inventors: VON GRAFENSTEIN, Ralf; Wasserbergs-
breite, D-37181 Hardegsen (DE). WOLTMANN, Sven;
Wasserbergsbreite, D-37181 Hardegsen (DE).

(74) Agent: LEVIN, Katrine, A.; Brown Raysman Millstein
Felder & Steiner LLP, 120 West 45th Street, New York, NY
10036 (US).

(51) International Patent Classification:
G06F 17/28

(54) Title: LANGUAGE TEACHING AND TRANSLATION SYSTEM AND METHOD

(57) Abstract: A computer-implemented system for translating words from one language to another is provided. The system allows a user to select a word for translation when the word appears in an application program or in an operating system on the computer. The system identifies words to be translated by copying and pasting the words from other window-based programs or by optically recognizing the characters of the words. The system uses a translation markup language to facilitate complex translations. The system provides teaching functionality to facilitate a user's learning of vocabulary words in multiple languages.

Published:
Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
LANGUAGE TEACHING AND TRANSLATION SYSTEM AND METHOD

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

The invention disclosed herein relates generally to systems and methods for translating words from a first language into a second language and for teaching vocabulary words in multiple languages to students. More particularly, the present invention relates to a computer-implemented system for automatically translating text from one language to another.

Modern computers are typically used for a number of purposes that include the creation, editing, viewing, and other use of textual information. Such computer applications as word processing, viewing World Wide Web pages, and scanning documents from hard copy into the computers’ memory may involve the use of textual information. With the growth of the Internet and the World Wide Web, more and more computer users are able to connect with ‘pages’ of information placed on the Web. Many of these Web pages contain textual information. Since the Web is a world-wide interconnection of computers and computer networks, there is no one language that is used for the textual content of all Web pages. In fact, information on the Web, in the form of content of Web pages, is available in a multitude of languages. Yet, it is the rare case that a particular Web page is available to viewers in more than one language. This often presents a problem for a user of the Web who wishes to view and understand the information contained on a Web page that is written in a language other than the user’s native or known languages. Likewise, computer users are often faced with the need to read, edit, and understand documents and work with programs written in another language, often in a language with which the computer user is personally unfamiliar, whether the document is a word processing document, a spreadsheet, a graph,
chart, or other textual information displayed by a computer, by an application program, or by an operating system, such as information presented to a user via menu bars, dialog boxes, etc. Likewise, there is an ever-increasing need for individuals to know foreign languages in order to facilitate communications with others around the world.

Thus there is a need for a computer-implemented system of translating words and phrases from one language to another and of teaching vocabulary words in multiple languages to students.

**BRIEF SUMMARY OF THE INVENTION**

It is an object of the present invention to provide a computer-implemented system that translates words from one language into another language.

It is another object of the present invention to provide a computer-implemented system that enables a user to translate the content of a World Wide Web page into a different language.

It is another object of the present invention to provide a computer-implemented system that translates words appearing on a computer screen from one language to another, whether the words appear as data in an application program, as with, for example, a word processing file or web page in a browser, or whether the words appear as part of the application program itself, as with, for example, menu bar options, dialog box choices, etc., in a graphical user interface (GUI) environment.

It is another object of the present invention to provide a computer-implemented system that translates words appearing on the computer screen as graphic images from one language to another.

It is another object of the present invention to provide a computer-implemented system that automatically updates a database of words and associated foreign-language translations so that the system may translate an ever-increasing number of words from one language into another.

It is another object of the present invention to provide a computer-implemented system that facilitates the teaching of vocabulary words in multiple languages to students.
The above and other objects are achieved by a computer-implemented system that translates words appearing on a computer screen from one language to another and updates a database of words and associated translations with words not previously stored in the database. Additionally, the system uses the database of words and associated translations to teach vocabulary words to students.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The invention is illustrated in the figures of the accompanying drawings which are meant to be exemplary and not limiting, in which like references refer to like or corresponding parts, and in which:

- Fig. 1 is a schematic representation of a computer system of a preferred embodiment of the present invention;
- Fig. 2 is a screen shot of a pop-up translator window of a preferred embodiment of the present invention;
- Fig. 3 is a listing of exemplary word class tags with corresponding English-language and German-language abbreviations and meanings of a preferred embodiment of the present invention;
- Fig. 4 is a listing of exemplary translation context tags with corresponding meanings and explanations of a preferred embodiment of the present invention;
- Fig. 5 is a listing of exemplary word form abbreviation tags with corresponding English-language and German-language representations and meanings of a preferred embodiment of the present invention;
- Fig. 6 is a listing of exemplary field abbreviation tags with corresponding English-language and German-language representations and meanings of a preferred embodiment of the present invention;
- Fig. 7 is a listing of exemplary general abbreviation tags with corresponding English-language and German-language representations of meanings of a preferred embodiment of the present invention;
- Fig. 8 is a schematic representation of a network-based computer system of a preferred embodiment of the present invention;
- Fig. 9 is an exemplary uniform resource locator;
Figs. 10A and 10B are flowcharts of the method for gathering data using the
WINDOWS edit control of a preferred embodiment of the present invention; and
Figs. 11A, 11B, 11C, 11D, and 11E are flowcharts of a method of optical
character recognition of a preferred embodiment of the present invention.

5 DETAILLED DESCRIPTION OF THE PREFERRED EMBODIMENTS

With reference to Fig. 1, one preferred embodiment of the present invention
includes a computer 100, having a random access memory (RAM) 110, a central processing
unit 120, one or more input devices 155, 160, and an output device, such as a display 150.
Preferably, the input devices include a keyboard 155 and a pointing device, such as a mouse
160. Preferably, the display 150 includes a CRT and/or LCD panel for presentation of data to
a user of the computer 100.

In a preferred embodiment of the present invention, the computer 100 runs an
operating system such as WINDOWS CE, WINDOWS 95, or WINDOWS 98, all available
from Microsoft Corporation, Redmond, WA, or other operating system capable of using a
15 graphical user interface. The computer may be a typical desktop computer, a hand-held
computer, an appliance with a built-in computer and/or microprocessor, or a specialized
computer, such as an MP3 audio player.

Typically, the computer is operated by loading one or more software programs
from one or more non-volatile sources 170 (e.g., floppy disks, a hard disk, read only memory
20 (ROM)) into the computer's RAM 110. The software is then executed by the computer 100.
In general, a visual rendition of the data occupying a given portion of the computer's memory
110 is displayed on the display 150. The data in the computer's memory 110 which is to be
displayed on the display 150 is generally one of two types of data: textual and graphic.
Textual data is commonly stored in the computer's RAM 110 as a series of numbers, with
each number being a code representing a particular character in an alphabet. A popular set of
codes to represent English-language-alphabet characters is the ASCII code, which assigns a
25 particular number to each English language character, as well as to a number of other
characters (e.g., "~", "!", "@", ",", "$", etc.). Each character of the English-language
alphabet (including both upper and lower case characters) is represented by a seven bit
number in the ASCII code. The complete ASCII code can be obtained from the American
National Standards Institute (ANSI), New York, NY. Thus, residing in a given portion of the computer's RAM 110 is a number which either represents a code for a particular character of a particular alphabet (as with the ASCII code for the English alphabet), or a number which instructs the computer to place a certain graphical image on a part of the display 150, to which that memory location is mapped.

If a given portion of the computer's RAM 110 is to be mapped to the display 150, the number at a given RAM location does not represent a code for a character, but rather represents instructions to the computer about what graphical image to display on the display 150. In such an instance, the data in the RAM location indicates to the computer the shape to be displayed on the display 150, as well as the color of the shape. Such data is termed 'graphics data' to distinguish it from the 'text data' which may also reside in the computer's RAM 110. Graphics data represent the color of each pixel of the display for which there is graphics data.

While the computer stores the text data and the graphics data differently in the computer's RAM 110, the display 150 viewable by a user of the computer 100 show equivalent images if the graphics data and the text data correspond. That is, it is possible to cause the display 150 to display a given character of a given alphabet in two ways. First, the ASCII or other code for the given character could be stored in the computer's RAM 110. Alternatively, stored in the computer's RAM 110 may be graphics data which instructs the computer how to form a given character by coding the color of each pixel of the display.

With reference to Figs. 1 and 2, a preferred embodiment of the system comprises a 'pop-up' translator window 200. The pop-up translator window 200 is activated by a user selecting for translation a given word or series of words that appear on the display 150. In a preferred embodiment, the user may select text for translation by using either the keyboard 155, such as by using cursor control keys to position a cursor over the text to be translated and then selecting the text by pressing a combination of keys (e.g., control+t), or by using the pointing device 160 and clicking-and-dragging the cursor to select the text to be translated.

Once the user has selected the text to be translated, the pop-up dictionary window 200 is displayed on the display 150. A preferred embodiment of the pop-up
dictionary window 200 is shown in Fig. 2. The pop-up dictionary window 200 shown in Fig.
2 may be opened on the display 150 in response to the user selecting the text "nutshell" on the
display for translation. As can be seen in Fig. 2, the pop-up dictionary window 200 includes
several elements. One of the elements included in pop-up dictionary window 200 is the text
that the user has selected for translation, in this case "Nutshell" 210. Next to the text 210
selected for translation is a translation of that text into another language 220, in this case
German ("Nusschale"). Along with the translation of the selected text, the pop-up dictionary
window 200 displays the type of speech of the translated word 230. In this case, the pop-up
dictionary window 200 displays an "f" to signify that the word ("nusschale") is a feminine
substantive noun in German. Below the translations of the word 220, of which there may be
several, the pop-up dictionary displays popular or common phrases 240 containing the
translated word in both the original and the translated languages. In the case shown in Fig. 2,
the pop-up dictionary displays the phrase "(to put it) in a ~ kurz gesagt" 240 to show the user
the translation of a common phrase which often incorporates the original translated word
("Nutshell" 210).

In a preferred embodiment of the system, the pop-up dictionary window 200
also includes four buttons that may be used by the user. A first button, the "close" button
250, represented by a circle with an 'X' in it, may be selected by the user to close the pop-up
dictionary window 200 and return to the task that the user was performing before opening the
pop-up dictionary window 200. For example, in a preferred mode of operation of the present
invention, if a user who had a large German vocabulary and a smaller English vocabulary
were using a web browser to view a World Wide Web page and came across the English
word "nutshell" and was unsure of its meaning, the user could select the word ("nutshell")
and open the pop-up dictionary window 200. The pop-up dictionary window 200 would open
on the display 150 and the user could read the German translation of the English word
("Nusschale"). Once the user is finished with the pop-up dictionary, the user could close the
window 200 by selecting the close button 250. This would close the window 200 in which
the pop-up dictionary appears, and return the user to the web browser to continue reviewing
the World Wide Web page.
A second button included in the pop-up dictionary window 200 of a preferred embodiment of the present invention is the “Speichern” button 260. The Speichern button 260 enables the user to save the word in the user’s personal dictionary. The personal dictionary is preferably stored in the computer’s non-volatile memory 170, such as a hard disk. The user’s personal dictionary is further described below.

A third button included in the pop-up dictionary window 200 of a preferred embodiment of the system is the “Lernbox” button 270. Pressing the Lernbox button 270 will cause the system to begin the “Lernassistent” program, as described below.

A fourth button included in the pop-up dictionary window 200 of a preferred embodiment of the system is the “Weiteres” button 280. When pressed, the Weiteres button 280 displays additional information about the selected word 210.

When the “Lernassistent” program is begun, such as by a user pressing the “Lernbox” button 270 in the pop-up window 200, the user will be provided with several options. One option will allow a user to select words contained in the main dictionary and copy the selected words into the user’s personal dictionary. The “Lernassistent” program generates virtual “index cards” for each of the words that a user adds to his or her personal dictionary. The user may make notes on the cards and may supplement this information with graphics and sounds. For example, a user may copy the word “cow” from the main dictionary into the user’s personal dictionary and place a clip-art graphic of a cow on the card to assist in remembering the definition of the word “cow”.

In a preferred embodiment of the present invention, the “Lernassistent” program enables a user to play educational, vocabulary-based games using the words stored in the user’s personal dictionary. For example, the “Lernassistent” program enables a user to play a simple word match program in which the English-language words stored in the user’s personal dictionary are displayed, one at a time and in a random order, to the user who must respond with the correct German-language word with the same meaning as the English-language word. Of course, the program may be set to operate in another mode wherein the German-language words are displayed to the user and the user must respond with the appropriate English-language word. While this example uses German and English as the two languages, the system can support any number of languages and both the main dictionary and
the personal dictionaries may contain words in multiple languages, all associated with common meanings.

In a preferred embodiment of the system, the words that are stored in the users' personal dictionaries are used by the system for several other purposes as well. Through "Learn-Channel" functionality, a company, individual, school, or other entity may select articles from the press, such as newspaper and magazine articles, and provide them to users of the system. This functionality is preferably used in conjunction with the Internet, such that the article source may transfer the article to an individual user's computer via the Internet. Such use of articles may be based on the specified interests and/or skills of an individual user, including language skills in the language of the articles. In addition to text articles, pictures, sounds, and videos may be transferred. With reference to Fig. 8, in a preferred embodiment of the system, the skills and interests of each user of the system are stored in a database 840 on a server computer 810. When supplied with an article for display, the system will display the article and highlight those words in the article that are in the user's personal dictionary. In a preferred embodiment of the present invention, the text of the articles will also be provided to the user in an audio format, such as MP3 or as a WAV file, such that the system may allow the user to hear the article being spoken.

Additionally, a preferred embodiment of the system allows the user to review and learn vocabulary. To do this, the system will present lessons and flash cards based on the user's personal dictionary.

The translation of a word is not limited to being a word in another language, but may be a complex explanation including words in the translated language and (mostly abbreviated) explanations in the source language. To store this explanation in a way that can be processed by a computer, a preferred embodiment of the system employs a language termed "Translation Markup Language" (TML) which is derived from the Extensible Markup Language (XML).

TML defines a text string containing regular text and defined tags which are used for structuring the TML string. A tag is represented as a defined string enclosed within the "<" and ">") characters.
The system arranges translations into 3 levels: meaning family; word class; and word meaning. A translation has one or more meaning families. In each meaning family, one or more word classes may be defined. In each word class, one or more word meanings may be defined. The meanings of these levels and how they are stored in TML are described below.

First, the translation of a word has one or more meaning families. A meaning family comprises two or more translated words in the destination language that are connected semantically in any way. An example is the word family that includes the word "fan," which may mean the type of fan that cools something (here it can either be a substantive noun or a verb, and has several different translations) and may also mean the type of fan who is interested in any kind of sport (in which case it is only a substantive noun and has a completely different translation than before). In TML, the tag <MFAMILY> indicates the beginning of the translation description of a meaning family.

Within a meaning family, one or more word classes may be defined. This is necessary since different translations are possible for different word classes. It is also possible that one translation is good for more than one word class. Word classes include, for example: adjectives, adverbs, articles, verbs, substantive nouns, etc. Word classes can be defined in more detail, for example: definite article, indefinite article, transitive verb, intransitive verb, reflexive verb, auxiliary verb, etc. In TML, the tag <WCLASS> indicates the beginning of the translation description of a word class or a word class group. This tag is followed by one or more "word class tags" that define the word class or classes for the following translation, followed by the translation description for that word class (group). Fig. 3 lists the word class tags 310 along with their associated English language abbreviations 320, English language meanings 330, German language abbreviations 340, and German language meanings 350.

Within a word class (group), one or more word meanings may be defined. This is necessary since different translations are possible for different word meanings. In TML, the tag <MEANING> indicates the beginning of the translation description of a word meaning. This tag is followed by the translation description, as defined below.
The translation for a single word meaning may still be very complex. TML may provide structure to the translation by using tags to: select a translation context; declare or define abbreviations; and declare or define fields. Fig. 4 lists the tags used to select a translation context 410, along with their associated meanings 420, and explanations 430. If no tag is specified or a tag is closed with the "</...>" tag, normal translation mode is activated.

Four kinds of abbreviations are used in a preferred embodiment of TML: word class abbreviations; word form abbreviations; general abbreviations; and field abbreviations.

Word class abbreviations 310 are specified behind the <WCLASS> tag to define the word class(es) of a translation, and are described above.

With reference to Fig. 5, word form abbreviations 510, including word forms like accusative, comparative, dative, infinitive, nominative, etc., can be specified in the TML string by pre-defined word form tags. Fig. 5 lists exemplary word form abbreviation tags 510, along with their associated English language representations 520, English language meanings 530, German language representations 540, and German language meanings 550. Word form abbreviation tags 510 may be entered as pre-defined abbreviation tags or as regular text and marked by the already defined tags "<expl>" and "</expl>" since they do not specify the translation, but rather are just for general explanation.

With reference to Fig. 6, field abbreviation tags 610 specify a field to which the word belongs, e.g. mathematics, physics, agriculture, etc. Fig. 6 lists exemplary field abbreviation tags 610 along with their respective English language representations 620, English language meanings 630, German language representations 640, and German language meanings 650.

With reference to Fig. 7, a preferred embodiment of the system uses predefined tags to specify general abbreviations. Fig. 7 lists sample exemplary general abbreviation tags 710, along with corresponding English-language representations 720 and meanings 730, as well as corresponding German-language representations 740 and meanings 750.

With reference to Fig. 8, in a network-based embodiment of the system, the system includes one or more client computers 800 used by users of the system, and at least
one server computer 810. The client computers 800 and the server computer 810 are operatively connected by connectors 820, such as the Internet, proprietary on-line networks such as America Online, intranets, or other known communication mechanisms. Residing on the server 810 are two databases 830, 840. One database, the dictionary database 830, comprises a general dictionary that translates words from one language to another and has related, dictionary-type information about each word stored with the words. The second database, the user database 840, stores information sorted by user, the information including words stored in each user's personal dictionary, skill levels of each of the users, lessons that each of the users has completed and associated scores on those lessons, as well as the general preferences of each of the users. Thus, the system may target certain materials, such as articles, to a given user based on the information about that user stored in the user database 840 on the server 810. Similarly, if a user attempts to translate a word that is not present in that user's personal dictionary, the system may send that word and appropriate other information from the user's computer 800 to the server 810. At the server 810, a system administrator may manually review the word and associated data and manually modify the dictionary database 830 at the server 810 to include the new word and its associated translations. The server may then be instructed to send the new word and its associated translations to each of the client computers 800 for them to update the personal dictionaries residing at each of the client computers 800.

In order for the pop-up dictionary to operate, it must know what text was selected by the user. Since the system does not receive the text as input directly, but rather is generally called upon to interpret text being used by another software program (such as in a web browser, a word processor, or an operating system), the system must be able to determine what text the user wants passed to the system’s pop-up dictionary for translation.

In a preferred embodiment of the system, there are two ways that the system may determine what text the user has selected. In many instances, the text that the user wants passed to the system is text that is stored in the computer’s memory as a string and is edited by the underlying software application (e.g., web browser, word processor, etc.) by using the WINDOWS edit control. With reference to Fig. 10A and 10B, once a user invokes the system 1010 and in order to determine whether the selected text resides in the WINDOWS
edit control, the system may query the operating system to identify the class of the window in
which the text appears 1020. The operating system will respond with a type of class, and the
system then determines whether that class is one from which, as with the WINDOWS edit
control, the system can copy and paste text 1030.

If the window in which the text appears is of a type (class) that supports
cutting and pasting, then the system can use the cut-and-paste functionality by sending the
target window a double-click message (using the SendMessage() procedure; this selects the
text appearing under the cursor) 1040 and then a “control-c” keyboard message (again using
the SendMessage() procedure; the control-c key combination is the WINDOWS system-wide
keyboard shortcut for copying selected text to the clipboard) 1050. Thus, the text under the
cursor is selected and copied to the clipboard 1060. Using the WINDOWS
GetClipboardData() procedure, the system may then paste the text from the clipboard into the
system 1070 for translation 1080 or other action. If in step 1030 the system determines that
the window’s class is not one which can be copied/pasted from, the system may invoke its
optical character recognition routines, as described below.

While this method works for those applications using the WINDOWS edit
control functionality, not every application program for which users will want to use the
present invention may have such functionality. Therefore, the system includes a second
method of determining what text data the user wishes to translate or otherwise have the
system use. In this method, the system provides optical character recognition (OCR) by
examining the text stored in memory as a map of pixels. With reference to Figs. 11A-E, first,
the OCR module must find an area on the screen which includes the word the user has
selected for translation by the system. Doing so requires the system to determine the color of
the word, determine the starting point (such as if the cursor is not on a pixel that comprises
one of the letters of the word), and then extend this starting point to a rectangle that covers all
the pixels that belong to the user-selected word.

Since the text is not necessarily black on a white background, the system must
determine the color of the selected text. After determining the position of the cursor 1110,
the system scans an area around the determined position of the cursor and the colors of all the
pixels are examined and totaled 1120. In a preferred embodiment of the present system, the
area around the determined cursor position is 11 pixels by 11 pixels. The color which occurs most often is assumed to be the background color. The color which occurs second most often is assumed to be the text color 1130. If the system continues with the OCR and is unsuccessful in determining the selected word, the system will start the OCR process over from this step 1130, reversing the assumptions about the background and text colors. That is, the system will then assume that the color which occurs most often is the text color and the color which occurs second most often is the background color.

Since the user may position the cursor on a pixel that is not part of the word (e.g., between two letters of the word; inside one letter of the word; slightly above or below the word; etc.) the system must scan the pixels around the cursor coordinates to ensure recognition of the proper word 1140. In doing so, a preferred embodiment of the system proceeds on a spiral path outward from the coordinates of the cursor to find a pixel that is the text color. Having found a pixel that is the text color, the system uses that pixel’s coordinates as the starting point for the next step 1140. The next step for the system is determining the boundaries of the selected word.

The system determines the boundaries of the entire selected word by beginning at the starting point established in the last step and expanding the area 1150 by, for example, one pixel to the top, one pixel to the bottom, and by two pixels to the left and two pixels to the right. This results in an area of, for example, 5 pixels by 3 pixels. Starting from the outside and working towards the center, all vertical and horizontal lines of pixels are checked to determine whether they contain at least one pixel in the text color 1160. If any one of the two vertical and two horizontal lines making up the outer perimeter of the area do not contain any pixel in the text color, that line is removed from the area 1170. The height of this area determines a temporary font size 1180. In a preferred embodiment of the system, this temporary font size is used to calculate a maximum distance \(d_{\text{max}}\) 1190 in terms of pixels between two letters with the following formula:

\[
d_{\text{max}} = \left(\text{temporary font size} / 3.2\right) + 0.9
\]

Using the value calculated for \(d_{\text{max}}\), the area is expanded by one pixel on each of the top and bottom and by \(d_{\text{max}}\) pixels to each of the left and right 1200. If the new area is not equal to the old area 1210, the process is then repeated from the step of checking all
vertical and horizontal lines from the outside to the inside for at least one pixel in the text color 1160. The process is stopped when no expansion of the area occurs 1210; that is, when the new area is equal to the old area. Such a situation indicates that there are no more letters that can be covered by the temporary expanded area.

Next, all vertical lines of this area are scanned from the left to the right. For each line, the system detects whether it has at least one pixel in the text color 1220. The state of each line is memorized until the next line has been scanned. There are four possible combinations of states of the last and the current line, as tabulated below, along with associated actions of a preferred embodiment of the system:

<table>
<thead>
<tr>
<th>Last line</th>
<th>Current line</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>empty</td>
<td>no action</td>
</tr>
<tr>
<td>empty</td>
<td>not empty</td>
<td>a new character starts; remember start line</td>
</tr>
<tr>
<td>not empty</td>
<td>empty</td>
<td>character has ended; call function described below</td>
</tr>
<tr>
<td>not empty</td>
<td>not empty</td>
<td>no action</td>
</tr>
</tbody>
</table>

Thus, the system first determines whether the current line has at least one pixel in the color of the word 1230. If it does not, the system next determines if the immediately preceding line had at least one pixel in the color of the word 1240. If it does, then the system continues to step 1250 and scales the character to the size of patterns in a pattern file, as discussed further below.

If in step 1230 the current line has at least one pixel in the color of the word, then the system next determines whether the immediately preceding line had at least one pixel in the color of the word 1260. If it did, the system moves one row to the right 1280 and continues the process at step 1230, as described above. If in step 1260 the immediately preceding line did not have at least one pixel in the color of the word, the system memorizes the current position as the start of a new character 1270, moves one row to the right 1280, and continues the process at step 1230, as described above. If in step 1240 the immediately
preceding line did not have at least one pixel in the color of the word, the system moves one row to the right 1280 and continues at step 1230, as described above.

In a preferred embodiment of the system, the basis for character recognition is a pattern file that contains the patterns of all characters in different fonts, font styles, and sizes, scaled to a fixed size of, for example, 16 pixels by 16 pixels. For recognizing a character, the pattern from the previous step is compared with every pattern in the pattern file. The most closely matching character is considered to be the recognized character.

The character pattern file comprises a number of character patterns and their corresponding ASCII codes. For example, each character pattern has a size of 16 pixels by 16 pixels and a gray scale color depth of 4. A pixel that completely belongs to the character is coded as 3 decimal (11 binary), a pixel that does not belong to the character at all is coded as 0 (00 binary), and pixels that are in between are coded as either 1 decimal (01 binary) or 2 decimal (10 binary).

Each character pattern file comprises not only one pattern but as many as needed for recognizing the character in different fonts, font styles, and font sizes. In general, there is not one pattern for each font and font size. When the character patterns are very similar, multiple font styles and sizes may be combined into one pattern by a logical binary OR operation.

Next, the system must find the character code (the ASCII code of the character) by searching the pattern file. First, the bitmap is scaled to the size of the patterns in the pattern file 1250, for example 16 pixels by 16 pixels. Next, all of the pixels that are the text color are converted to 1 while all the pixels that are not the text color are converted to zero. This makes the bitmap compatible with the pattern file.

Next, the system compares the scaled character with every pattern in the pattern file. A matching score is calculated for each character in the pattern file using the following procedure. Initially, the matching score is set to zero 1290. A pattern is selected from the pattern file for comparison with the character 1300. For each pixel of the pattern, the system calculates the difference of the color value of that pixel in the character to be recognized and the color value of the corresponding pixel in the pattern file 1310. If this difference is less than zero, meaning that there is a pixel in the character where there is not
one in the pattern file character, the difference is multiplied by, for example, (–14 decimal) and added to the matching score 1320. If the difference is equal to or greater than zero, the difference is added to the matching score 1320. This process is repeated for each pixel in the character 1310, 1320. The process is repeated from step 1290 until all patterns in the pattern matching file have been compared with the character 1330. The character in the pattern file with the highest matching score is determined to be the detected character 1340. This process is then repeated for each character in the selected word 1350. Once all of the characters have been compared, the OCR process ends 1360.

Many client-server applications require binary data to be transported through the Internet bi-directionally. Since many companies and other entities use firewalls or other devices to secure their Internet connections and often allow only HTTP traffic to pass the firewalls. Further, HTTP-based clients may be limited in their ability to send binary data.

The HTML protocol establishes a standardized way for web browsers to submit data that a user may enter on a form. One of the methods that can be used is to do so with the known GET-method. When submitting data with the GET-method, a web browser calls the page that receives the data with a uniform resource locator (URL) in the format shown in Fig. 9. The first part of this URL, “http://” 910 specifies the protocol to use; in this case, the hypertext transfer protocol (HTTP). The next part, “hostname.dom” 920 specifies the domain name system (DNS) host name of the server computer. The “pagename” 930 specifies the name of the HTML document or script that the client is asking for. After the slash “/”, name/value pairs may be specified, with each pair separated by ampersands (“&”), such as represented by characters 940.

Normally, all characters except for a specific set of standard characters that are allowed in the name/value pairs are encoded in the form: %ASCII code in hexadecimal].

As an example, a white space character has the decimal ASCII code of 32, which is coded as 20 in hexadecimal. Thus, the string “Hello World” would be encoded as “Hello%20World”.

When transporting binary data, most of the characters are non-valid characters for URL’s and therefore need to be written in the format specified above, making transporting binary data across firewalls very inefficient.
However, it has been discovered that by using the well-known technique of Huffman-encoding, the binary transport of data in this way is greatly enhanced. Using the Huffman technique, the data to be sent is encoded by converting 8-bit bytes into data words of different lengths. Bytes that occur less often are encoded as longer codes and bytes that occur more often are encoded as smaller codes.

Since a server may receive requests from more than one client at a time and since the order in which the packages will be received is unknown, each transfer that has more than one package of data must be assigned a unique transfer identifier.

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made without departing from the spirit and scope of the invention, and the invention is thus not to be limited to the precise details of methodology or construction set forth above as such variations and modification are intended to be included within the scope of the invention.
WHAT IS CLAIMED IS:

1. A system for translating text from a first language to a second language, characterized by a first dictionary database (830), stored on a first computer (810), the first dictionary database associating words in a first language with words in a second language; a second dictionary database, stored on a second computer (800), the second dictionary database associating words in the first language with words in the second language; and translating software for translating a word selected by a user in the first language into the second language, which software determines whether the selected word in the first language is contained in the second dictionary database and, if not, queries the first dictionary database (830) for words in the second language that have the same meanings as the selected word in the first language.

2. The system of claim 1, wherein the first computer (810) and the second computer (800) are each connected to a computer network (820), e.g. the Internet, and wherein the first computer (810) is a server and the second computer (800) is a client of the network.

3. The system of claim 1 or 2, characterized by a teaching software for selecting words in a first language from the first or second dictionary database, for presenting the selected words, one at a time, to a user, for receiving an input from the user responsive to each word, and for determining whether the input from the user means a correct translation of the respective word into the second language by utilizing a translation produced by the translation software.

4. The system of claim 1, 2 or 3, wherein a user preferences database (840) is stored on the first computer (810), the user preferences database (840) comprising user data corresponding to interests and skill levels of users of the system and wherein the translating and/or teaching software queries the user preferences database (840) to obtain the preferences and skill levels of a user of the system.

5. The system of claim 4, wherein the translating and/or teaching software updates the user data stored in the user preferences database (840) based on how the users of the system use the system.
6. The system of claim 4 or 5, wherein the teaching software determines which words to present to the user based on the preferences and skill levels of the user.

7. The system of claim 3, 4, 5 or 6, wherein the teaching software presents the words to the user in the form of virtual index cards.

8. The system of claim 3, 4, 5, 6 or 7, wherein the teaching software receives an input from the user indicating a data file to store with an associated word and wherein the software associates the data file with a word in the second dictionary database.

9. The system of claim 1, 2, 4 or 5, wherein the translating software copies the selected word in the first language from another computer program executing on the second computer (800) to a temporary data storage area and pastes the selected word from the temporary data storage area into the translating software.

10. The system of claim 9, wherein a computer operating system executing on the second computer (800) controls data copied to and pasted from the temporary data storage area, and wherein the translating software copies and/or pastes the selected word by sending the computer operating system a message emulating an input from a user of the system.

11. A method of optical character recognition for determining a word selected by a user, preferably for determining the selected word in a system according to claim 1, 2, 4 or 5, the word being represented by a plurality of pixels displayed on a display, characterized by the steps of:

   (1) calculating a starting point based on an input from the user;

   (2) determining an area having within its perimeter substantially all of the plurality of pixels representing the word;

   (3) determining a number of individual letters the word comprises;

   (4) determining which of the plurality of pixels representing the word each of the individual letters the word comprises;

   (5) determining an identity of each of the individual letters the word comprises, preferably by comparing with character patterns (Fig. 11).

12. The method of claim 11, wherein the pixels representing the word are of a first color and wherein step (2) comprises the steps of:

   (a) calculating the boundaries of an area around the starting point;
(b) determining a color of each pixel displayed on the display within the area around the starting point;

(c) determining the first color to be a second-most prevalent color of pixels displayed on the display within the area around the starting point;

(d) starting at a first pixel of the first color, calculating the boundaries of an area surrounding the first pixel of the first color;

(e) starting with pixels along the boundaries of the area surrounding the first pixel of the first color and working inward, determining whether each vertical and horizontal line of pixels contain at least one pixel of the first color and, if any one of each vertical and horizontal lines do not contain at least one pixel of the first color, removing that line of pixels from the area surrounding the first pixel in the first color;

(f) calculating a temporary font size to be the height in pixels of the area surrounding the first pixel in the first color;

(g) calculating a maximum distance between letters based on the temporary font size;

(h) calculating an expanded area surrounding the first pixel in the first color by adding one row of pixels to each of the top and bottom of the area surrounding the first pixel in the first color and adding a number of rows of pixels to each of the left and right of the area surrounding the first pixel in the first color equal to the maximum distance between letters;

(i) repeating steps (c) through (h) until the expanded area surrounding the first pixel in the first color is equal to the area surrounding the first pixel in the first color (Fig. 11).

13. A system according to anyone of claims 1-10 or according to the entering clause of claim 1, wherein a translation markup language is used, comprising meaning families; word classes; and word meanings (Fig. 3, 4 and 5).

14. The system of claim 13, wherein the meaning families, word classes, and word meanings are indicated by tags.

15. The system of claim 13 or 14, wherein translation contexts are indicated by translation context tags.
FIG. 1
FIG. 2

Cleverlearn.com

→ Nutshell

1. Nusschale

(to put it) in a - kurz gesagt

2. 230

3. 240

4. 250

5. 260

6. 270

7. 280

8. Spielchen

9. Lernbox

10. Weiteres
<table>
<thead>
<tr>
<th>tag</th>
<th>abbr. English</th>
<th>meaning English</th>
<th>abbr. German</th>
<th>meaning German</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;adj&gt;</td>
<td>adj</td>
<td>adjective</td>
<td>Adj</td>
<td>Adjektiv, Eigenschaftswort</td>
</tr>
<tr>
<td>&lt;adv&gt;</td>
<td>adv</td>
<td>adverb</td>
<td>Adv</td>
<td>Adverb, Umstandswoart</td>
</tr>
<tr>
<td>&lt;adv/interrog&gt;</td>
<td>interrog adv</td>
<td>interrogative adverb</td>
<td>Interrog Adv</td>
<td>fragendes Adverb &quot;wozu&quot;, &quot;wovor&quot;</td>
</tr>
<tr>
<td>&lt;adv/cj&gt;</td>
<td>cj adv</td>
<td>conjunctional adverb</td>
<td>Konj Adv</td>
<td>Konjunktionalsadjverb</td>
</tr>
<tr>
<td>&lt;art&gt;</td>
<td>art</td>
<td>article</td>
<td>Art</td>
<td>Artikel</td>
</tr>
<tr>
<td>&lt;art/det&gt;</td>
<td>det art</td>
<td>definite article</td>
<td>best Art</td>
<td>bestimmter Artikel (der, die, das, ...)</td>
</tr>
<tr>
<td>&lt;art/indet&gt;</td>
<td>indet art</td>
<td>indefinite article</td>
<td>unb Art</td>
<td>unbestimmter Artikel (ein, eine, ...)</td>
</tr>
<tr>
<td>&lt;cj&gt;</td>
<td>cj</td>
<td>conjunction</td>
<td>Konj</td>
<td>Konjunktion, Bindewort</td>
</tr>
<tr>
<td>&lt;int&gt;</td>
<td>int</td>
<td>interjection</td>
<td>Int</td>
<td>Interjektion, Ausruf</td>
</tr>
<tr>
<td>&lt;part&gt;</td>
<td>part</td>
<td>particle</td>
<td>Part</td>
<td>Partikel</td>
</tr>
<tr>
<td>&lt;prep&gt;</td>
<td>prep</td>
<td>preposition</td>
<td>PrP</td>
<td>Präposition</td>
</tr>
<tr>
<td>&lt;pron&gt;</td>
<td>pron</td>
<td>pronoun</td>
<td>Pron</td>
<td>Pronomen</td>
</tr>
<tr>
<td>&lt;pron/dem&gt;</td>
<td>dem pron</td>
<td>demonstrative pronoun</td>
<td>Dem Pron</td>
<td>Demonstrativpronomen (hinw. Pron.,)</td>
</tr>
<tr>
<td>&lt;pron/indet&gt;</td>
<td>indet pron</td>
<td>indefinite pronoun</td>
<td>unb Pron</td>
<td>Unbestimmtes Pronomen (irgendw.)</td>
</tr>
<tr>
<td>&lt;pron/interrog&gt;</td>
<td>interrog pron</td>
<td>interrogative pronoun</td>
<td>Interrog Pron</td>
<td>Interrogativpronomen (Fragewort)</td>
</tr>
<tr>
<td>&lt;pron/pers&gt;</td>
<td>pers pron</td>
<td>personal pronoun</td>
<td>Pers Pron</td>
<td>Personalpronomen</td>
</tr>
<tr>
<td>&lt;pron/rel&gt;</td>
<td>rel pron</td>
<td>relative pronoun</td>
<td>Rel Pron</td>
<td>Relativpronomen</td>
</tr>
<tr>
<td>&lt;pron/poss&gt;</td>
<td>poss pron</td>
<td>possessive pronoun</td>
<td>Poss Pron</td>
<td>Possessivpronomen (sein, mein, ihr)</td>
</tr>
<tr>
<td>&lt;pron/refl&gt;</td>
<td>refl pron</td>
<td>reflexive pronoun</td>
<td>Refl Pron</td>
<td>Reflexivpronomen (sich, mich, ...)</td>
</tr>
<tr>
<td>&lt;s&gt;</td>
<td>s</td>
<td>substantive</td>
<td>S</td>
<td>Substantiv</td>
</tr>
<tr>
<td>&lt;s/f&gt;</td>
<td>f</td>
<td>feminine substantive</td>
<td>f</td>
<td>feminines Substantiv</td>
</tr>
<tr>
<td>&lt;s/m&gt;</td>
<td>m</td>
<td>masculine substantive</td>
<td>m</td>
<td>maskulines Substantiv</td>
</tr>
<tr>
<td>&lt;s/n&gt;</td>
<td>n</td>
<td>neuter substantive</td>
<td>n</td>
<td>neutrales Substantiv</td>
</tr>
<tr>
<td>&lt;s/pl&gt;</td>
<td>pl</td>
<td>plural substantive</td>
<td>pl</td>
<td>pluralistisches Substantiv</td>
</tr>
<tr>
<td>&lt;s/plf&gt;</td>
<td>pl f</td>
<td>pl. feminine substantive</td>
<td>pl f</td>
<td>pl. feminines Substantiv</td>
</tr>
<tr>
<td>&lt;s/plm&gt;</td>
<td>pl m</td>
<td>pl. masculine substantive</td>
<td>pl m</td>
<td>pl. maskulines Substantiv</td>
</tr>
<tr>
<td>&lt;s/pln&gt;</td>
<td>pl n</td>
<td>pl. neuter substantive</td>
<td>pl n</td>
<td>pl. neutrales Substantiv</td>
</tr>
<tr>
<td>&lt;v&gt;</td>
<td>verb</td>
<td>verb</td>
<td>V</td>
<td>Verb</td>
</tr>
<tr>
<td>&lt;v/aux&gt;</td>
<td>aux</td>
<td>auxiliary verb</td>
<td>Vhilf</td>
<td>Hilfsverb</td>
</tr>
<tr>
<td>&lt;v/aux&gt;</td>
<td>aux</td>
<td>negated auxiliary verb</td>
<td>Vihilf</td>
<td>verneinutes Hilfsverb</td>
</tr>
<tr>
<td>&lt;v/&gt;</td>
<td>v</td>
<td>intransitive verb</td>
<td>V</td>
<td>Intransitives Verb</td>
</tr>
<tr>
<td>&lt;v/impers&gt;</td>
<td>impers</td>
<td>impersonal verb</td>
<td>V/unpers</td>
<td>unpersoenliches Verb</td>
</tr>
<tr>
<td>&lt;v/&gt;</td>
<td>v</td>
<td>reflexive verb</td>
<td>Vr</td>
<td>reflexives Verb</td>
</tr>
<tr>
<td>&lt;v/&gt;</td>
<td>v</td>
<td>transitive verb</td>
<td>Vt</td>
<td>transitives Verb</td>
</tr>
</tbody>
</table>

**Fig. 3**
<table>
<thead>
<tr>
<th>Tag</th>
<th>Meaning</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;expl&gt;</code></td>
<td>explanation</td>
<td>an explanation in the <em>either</em> language</td>
</tr>
<tr>
<td><code>&lt;ref&gt;</code></td>
<td>reference</td>
<td>referencing words of the <em>source</em> language</td>
</tr>
<tr>
<td><code>&lt;syn&gt;</code></td>
<td>synonym</td>
<td>synonym in the <em>source</em> language</td>
</tr>
<tr>
<td><code>&lt;comb&gt;</code></td>
<td>combination</td>
<td>word combination in the <em>source</em> language</td>
</tr>
<tr>
<td><code>&lt;prep_transl&gt;</code></td>
<td>preposition translation</td>
<td>translation of a preposition used with the word <em>source</em> language <strong>&lt;transl&gt; destination language</strong></td>
</tr>
<tr>
<td>Tag</td>
<td>repres. English</td>
<td>meaning English</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>adm</td>
<td>adm.</td>
<td>administrative</td>
</tr>
<tr>
<td>agr</td>
<td>agr.</td>
<td>agriculture</td>
</tr>
<tr>
<td>anat</td>
<td>anat.</td>
<td>anatomy</td>
</tr>
<tr>
<td>antiqu</td>
<td>antiqu.</td>
<td>antiquity</td>
</tr>
<tr>
<td>arch</td>
<td>arch.</td>
<td>architecture</td>
</tr>
<tr>
<td>astr</td>
<td>astr.</td>
<td>astronomy, astrology</td>
</tr>
<tr>
<td>avi</td>
<td>avi.</td>
<td>aviation</td>
</tr>
<tr>
<td>bibl</td>
<td>bibl.</td>
<td>biblical</td>
</tr>
<tr>
<td>biol</td>
<td>biol.</td>
<td>biology</td>
</tr>
<tr>
<td>bot</td>
<td>bot.</td>
<td>botany</td>
</tr>
<tr>
<td>chem</td>
<td>chem.</td>
<td>chemistry</td>
</tr>
</tbody>
</table>

**FIG. 6**
<table>
<thead>
<tr>
<th>tag</th>
<th>repred. English</th>
<th>meaning English</th>
<th>repres. German</th>
<th>meaning German</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a.</td>
<td>also</td>
<td>a.</td>
<td>auch</td>
</tr>
<tr>
<td>Am</td>
<td>Am.</td>
<td>American English</td>
<td>Am.</td>
<td>amerikanisches Englisch</td>
</tr>
<tr>
<td>amer</td>
<td>amer.</td>
<td>american</td>
<td>amer.</td>
<td>amerikanisch</td>
</tr>
<tr>
<td>abbr</td>
<td>abbr. of</td>
<td>abbreviation of</td>
<td>Abb. von</td>
<td>Abkürzung von</td>
</tr>
<tr>
<td>comm</td>
<td>comm.</td>
<td>commonly</td>
<td>allg.</td>
<td>allgemein</td>
</tr>
<tr>
<td>Br</td>
<td>Br.</td>
<td>British English</td>
<td>Br.</td>
<td>britisches Englisch</td>
</tr>
<tr>
<td>bs</td>
<td>b.s.</td>
<td>bad sense</td>
<td>schl. S.</td>
<td>in schlechten Sinne</td>
</tr>
<tr>
<td>esp.</td>
<td>esp.</td>
<td>especially, particularly</td>
<td>bsd.</td>
<td>besonders</td>
</tr>
<tr>
<td>etc</td>
<td>etc.</td>
<td>etcetera</td>
<td>usw.</td>
<td>usw.</td>
</tr>
<tr>
<td>coll</td>
<td>coll.</td>
<td>collective noun</td>
<td>Koll.</td>
<td>Kollektivum, als Sammelwort</td>
</tr>
</tbody>
</table>

[...and so on...]
FIG. 8
http://hostname.dom/page/name/name&value

910 920 930 940

Fig. 9
1010
USER INVOKES
TRANSLATOR SYSTEM

1020
TRANSLATOR SYSTEM
QUERIES OPERATING
SYSTEM TO DETERMINE
THE CLASS OF THE
WINDOW IN WHICH THE
TEXT RESIDES

1030
IS CLASS ONE
WHICH CAN BE
COPIED/PASSED
FROM?

N
GO TO OCR
PROCEDURE

1040
Y
SEND WINDOW WITH
TEXT A "SELECT"
MESSAGE TO SELECT
THE TEXT

1050
SEND WINDOW WITH
TEXT A "COPY"
MESSAGE TO COPY
THE TEXT

FIG. 10A
THE TEXT IS COPIED, SUCH AS TO THE WINDOWS CLIPBOARD

PASTE COPIED TEXT, SUCH AS FROM THE WINDOWS CLIPBOARD TO TRANSLATOR SYSTEM

OPEN POP-UP DICTIONARY WINDOW WITH TEXT AND TRANSLATION

1060

1070

1080
DETERMINE CURSOR POSITION

SCAN AREA AROUND CURSOR POSITION, EXAMINE AND COUNT COLORS OF PIXELS IN AREA SCANNED

DETERMINE SECOND-MOST-OFTEN OCCURRING COLOR TO BE COLOR OF PIXELS COMPRISING WORD

DETERMINE STARTING POINT BY SCANNING AROUND CURSOR POSITION FOR PIXEL OF SECOND-MOST-OFTEN COLOR

A

Fig. 11A
EXPAND AREA VERTICALLY AND HORIZONTALLY

STARTING FROM THE OUTSIDE AND WORKING TOWARDS THE STARTING POINT, CHECK EACH HORIZONTAL AND VERTICAL ROW FOR A PIXEL IN THE COLOR OF THE WORD

ELIMINATE FROM THE AREA ALL PERIMETER BOWS THAT DO NOT CONTAIN ANY PIXELS IN THE COLOR OF THE WORD

CALCULATE TEMPORARY FONT SIZE AS HEIGHT OF AREA

FIG 11B
CALCULATE MAXIMUM DISTANCE BETWEEN TWO LETTERS, \( d_{\text{max}} \)

CALCULATE NEW AREA BY EXPANDING AREA BY ONE PIXEL ON TOP AND ON BOTTOM AND BY \( d_{\text{max}} \) PIXELS ON LEFT AND RIGHT

IS NEW AREA EQUAL TO AREA?

WORKING FROM LEFT TO RIGHT, SCAN EACH VERTICAL LINE FOR AT LEAST ONE PIXEL IN THE COLOR OF THE WORD

FIG. 11C
DOES THE CURRENT LINE HAVE AT LEAST ONE PIXEL IN THE COLOR OF THE WORD?

DID THE IMMEDIATELY RECEDING LINE HAVE AT LEAST ONE PIXEL IN THE COLOR OF THE WORD?

SCALE CHARACTER TO SIZE OF PATTERNS IN PATTERN FILE

MOVE ONE ROW TO THE RIGHT

MEMORIZE POSITION AS START OF NEW CHARACTER

FIG. 12D
E

SET MATCHING SCORE TO ZERO

SELECT PATTERN FROM PATTERN FILE FOR COMPARISON WITH CHARACTER

CALCULATE DIFFERENCE BETWEEN COLOR VALUE OF PIXEL IN CHARACTER WITH COLOR VALUE OF PIXEL IN PATTERN FOR EACH PIXEL IN CHARACTER

IF DIFFERENCE IS >0, MULTIPLY BY (14) AND ADD TO MATCHING SCORE, OTHERWISE ADD DIFFERENCE TO MATCHING SCORE

HAVE ALL PATTERNS BEEN COMPARED?

Y
ASSUME CHARACTER CORRESPONDS TO PATTERN WITH HIGHEST MATCHING SCORE

MORE CHARACTERS IN WORD?

N
END OCR

N

FIG. 11E