Enhancing Language Learning and Translation with Ubiquitous Applications

Bartholomäus Wloka and Werner Winiwarter
University of Vienna
Department of Scientific Computing
Universitätsstraße 5, 1010 Vienna, Austria
{bartholomaeus.wloka,werner.winiwarter}@univie.ac.at

ABSTRACT

In this paper we introduce a comprehensive framework for a ubiquitous translation and language learning environment utilizing the capabilities of modern cell phone technology. We present a partial first realization of this framework: an application for learning Japanese characters and Japanese-English translation; the latter is based on results from our previous work. For the implementation, we have used the open-source Maemo platform on the Nokia N900, a state-of-the-art mobile device. We present the architecture of our framework, our current state of implementation, and the findings we have gathered so far.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Machine Translation*; J.5 [**Computer Applications**]: Arts and Humanities—*Language Translation*; K.3.1 [**Computers and Education**]: Computer Uses in Education—*Computer-Assisted Instruction, Distance Learning*

General Terms

Algorithms, Design

Keywords

Ubiquitous language learning, ubiquitous machine translation, mobile applications, mobile devices

1. INTRODUCTION

With the growing usage of mobile devices, their increasing computational power, and an improving network infrastructure, the concept of mobile and ubiquitous computing is becoming an increasingly important part of our everyday life. Consequently, there is a growing demand for mobile applications to support our daily life activities and provide various forms of entertainment. One of the most needed application types is translation and language learning software. Even though there exist Web-based applications and a recently growing number of cell phone applications offering these services, there is often a limit to their usefulness in an everyday setting. Despite the capabilities of todays cell phones, they rarely utilize the sensors' capabilities of cell phones, e.g. to adjust the learning material difficulty to the noise level, which affects the ability of the student to concentrate, or to consider the GPS position to offer situation- and location-based learning suggestions. In other words, current solutions lack the situational awareness and thereby important aspects of *ubiquitousness*.

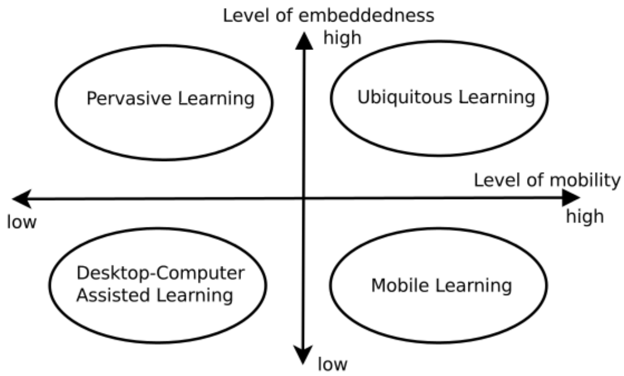

Figure 1: Types of learning environments [14]

According to [14], Fig. 1 demonstrates the correlation of the important terms in a computer-assisted learning context. Computer assisted learning is usually software or Web content accessible from a local workstation. The easier it is for the students to carry around this program and access it wherever they are, e.g. utilizing a PDA, the more it becomes a *mobile learning* application. The *pervasive* component is added if the application and the device are also able to measure and/or adjust to the environment of the students. A combination of pervasive and mobile learning is called *ubiquitous learning* and is defined by a high degree of *embeddedness* and *mobility*. Similarly, we define a ubiquitous translation system as a mobile translation system, which uses information about the current situation to refine or expand the translation, make suggestions for useful terms, sentences, names, etc.

In order to transform this theory into a real application, a proper platform is needed: a device which offers portability, computational power, sensory capabilities, and ease of use. The modern cell phone meets all those prerequisites and is, beyond that, already widespread. As Mark Weiser puts it:

MoMM2010, 8-10 November, 2010, Paris, France.

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [24]

Cellular phones are on the verge of becoming such a profound technology. The usability of cell phones is improving through increasing network coverage, growing wireless telecommunication capabilities, and enhanced battery technology. The application range presented to us by our small hand-held devices is expanding tremendously. Cell phones with integrated browsers and GPS localization are no exception any more and their applications are becoming more user friendly and more powerful. The recent merge of internet tablets and cell phones undertaken by Nokia, in the form of the *Nokia N900* is a good example of a big step towards a device which can handle communication and computation tasks of a desktop computer, while maintaining the size and capabilities of a regular cell phone. This enables us to access information from wherever we are and whenever we want and process this information, similar as on a desktop computer. However, compared with a desktop computer we have less hardware resources but additional sensors attached to it, such as GPS position calculation, noise levels measured by the microphone, and acceleration data measured by the phone's movement sensors. Furthermore, there is a smaller screen to display all this information to the user. Hence, we have to put much effort into the design of the user interface and the architecture regarding computation times. The primary reason why we have decided to develop on the Nokia N900 is the fact that its native operating system, Maemo5, is a Linux-based distribution and amongst other advantages, offers the ability to use *SCIM* (Smart Common Input Method). This enables the input of Japanese characters, in analogy to the standard input method in Japan, when utilizing western-style keyboards.

In this paper we present our research on how a cell phone can be utilized and how its capabilities can be exploited to provide the best possible language learning and translation services, leading to a ubiquitous language learning and translation environment. We report on our implementation experience, which we have gathered so far, and describe some of our previous research, the *TRanslation Enhancement Framework* (TREF), which we have integrated into the current work to create the *Ubiquitous TRanslatiOn and Language Learning* environment (UTROLL). In Sect. 5, a showcase narrates the details of the Japanese-English translation module and the *kanji* (Japanese characters) learning module, which we have implemented as part of UTROLL.

Throughout this paper we present examples for the language pair Japanese–English, as we focus on these languages in our research work. Nonetheless, the system is modular and flexible, and an adjustment or extension to other languages would be a matter of changing mere implementation details and adding the language-specific resources, such as lexica, parser, corpora, etc. It is important to mention, however, that our translation framework is specifically designed and well-suited for languages with radically different surface characteristics, e.g. European vs. Asian languages.

The rest of this paper is organized as follows. In Sect. 2 we discuss research relevant to our work. We then describe the system architecture of our translation and language learning environment in Sect. 3. We elaborate on our earlier research efforts, which we have integrated into this work, in Sect. 4, and in Sect. 5 we present the showcase application to give the reader a tangible impression of the already implemented components. We conclude the paper with a summary and an outlook in Sect. 6.

2. RELATED WORK

There is a variety of Web-based translation services and language learning applications available, e.g. [8, 29, 16]. The translation services adopt techniques from machine translation, which has ever since included many different approaches. An overview of those can be obtained from [26]. In our previous research [30], we have shown that creating templates from sequence-aligned [11] and clustered sentence pairs, which were found to be similar, is a feasible way of improving state-of-the-art statistical machine translation. Based on [29, 28, 27] we have found that our method, being *example* and *corpus-based*, is well suited for a language learning application, since it contains intermediate information in the translation processes, which can be valuable for a language student.

The ubiquitous property of mobile devices has been successfully used to offer contextual learning environments in the past [31, 19, 7]. One of the implementations is a system by [20], where the learners could utilize PDAs to find the right politeness form while formulating a Japanese sentence in a certain situation. The application of ubiquitous learning demands several characteristics to be fulfilled [5]:

- *Permanency*: Learners never lose their work unless it is purposefully deleted. In addition, the entire learning process is recorded continuously.
- *Accessibility*: Learners have access to their documents, data, or videos from anywhere. That information is provided based on their requests. Therefore, the learning involved is self-directed.
- *Immediacy*: Wherever learners are, they can get any information immediately. Thus, learners can solve problems quickly. Otherwise, the learner can record the question and look for the answer later.
- *Interactivity*: Learners can interact with experts, teachers, or peers in the form of synchronous or asynchronous communication. Hence, the experts are more reachable and the knowledge becomes more available.
- *Situation of instructional activities*: The learning could be embedded in our daily life. The problems encountered as well as the knowledge required are all presented in their natural and authentic forms. This helps learners to notice the features of problem situations that make particular actions relevant.

An example of the application of those guidelines can be seen in [1]. The idea behind it is to give the students the chance to efficiently use their time and the ability to access class room information at will. In [10] a ubiquitous learning environment was developed by using IEEE 802.11 WLAN and Bluetooth for network communication. This showed that a learning experience, supported by a contextually matching surrounding, is more valuable in terms of understanding and memorizing since it is based on an inductive process. However, the limits of the network technologies do not allow a deployment of that system into a large network structure.

As pointed out by [18], language learning is a life-long activity, and support by ubiquitous learning environments can accompany the learner at all stages. Language learning takes place virtually anywhere and is optimized if supplemented on demand. The need for immediate help makes translation resources on mobile devices very valuable, which is the reason we have focused on combining those two, so that they complement each other in the best way possible.

In the research area of intelligent tutoring [21], it was pointed out that the use of multiple representations delivers better results than, for example, only repetition. The use of pictures as representations of objects, which were in the focus of the tutoring applications, proved to be very useful and enhanced the learning experience significantly. We have applied this idea in our work and present the learning contents graphically, as well as through several different representations. Furthermore, there is need for an efficient and clear user interface design [1]. This is a vital point, considering the vast amount of information available on a mobile device, combined with limited output capabilities, e.g. the small display. [4] has addressed this issue in the CAMELEON project, proposing a dynamic user interface.

According to the *cognitive load theory* [22], the amount of information presented to the learner has a critical effect on the outcome of the learning process. Therefore, it is an important design consideration while building a learning environment. Only information which is stored by the short term memory can be processed and then stored in long term memory. If the student is faced with too much information at once, some of it might never be properly processed and is therefore lost. Too little information can lead to an "idle" state of the short term memory, which makes it less productive. Hence it is vital to keep adequate levels of information presentation, depending on the students' abilities as well as the situation. The abilities are defined by how advanced they are in the learning process, which can be monitored by short quizzes. The situation, in this context, is monitored by sensors, such as GPS and microphone, which measure how well the students can focus on their study at any given time. Research efforts by [6] show that using contextual information, i.e. information inferred from the location and the activities should be associated with the learning context to present a meaningful learning content.

3. SYSTEM ARCHITECTURE

The framework is designed as a client-server setup. The client is the cell phone, in our case the Nokia N900. The LAMP (Linux-Apache-MySQL-PHP) server is situated at the University of Vienna. The hardware specifications of both are listed in Table 1. The operating system on the N900 is the open-source Maemo5, running on a Linux kernel, designed for smartphones and Internet tablets. This platform was developed in collaboration with the Linux kernel, Debian, and the GNOME project. The GUI and application framework is Hildon, which was originally developed by Nokia and became recently a part of GNOME. For development we have used Scratchbox, a cross-compilation toolkit (`www.scratchbox.org/`), in combination with Python and GTK+, a cross-platform widget toolkit for creating graphical user interfaces. In contrast to distributions for desktop computers, Maemo5 has touch screen support, sliding keyboard support, an interface to the camera and to other sensors, while discarding some typical desktop distri-

	Server	Client - Nokia N900
CPU:	4x Intel(R) Xeon(R) CPU E5405 @ 2.00GHz	ARM Cortex A8 600 MHz, PowerVR SGX graphics
Mem:	4057MB	256MB
OS:	Ubuntu 9.10	Maemo5
Ver:	Linux 2.6.31-17-server (x86_64)	Nokia-N900-51-1 Linux 2.6.28-omap1 on armv7l
HD:	1.7 TB	32 GB

Table 1: Hardware specifications

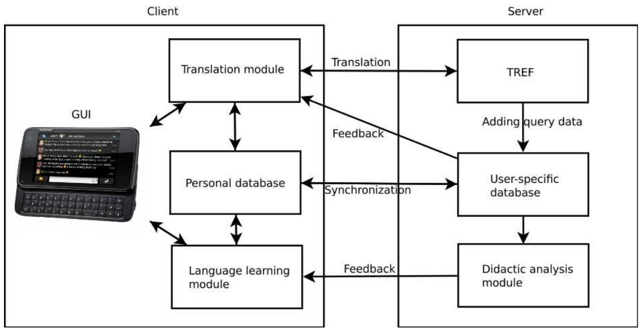

Figure 2: System architecture

bution functionalities. Even though the hardware on the client side is quite powerful for a cell phone, it is not enough to perform all the calculations needed for our framework in a reasonable amount of time. Hence, we need to outsource as many calculations as possible to the server, where they can be processed much quicker. Another reason is the fact that the main part of the translation module, the *TRanslation Enhancement Framework* (TREF), utilizes SWI-Prolog [25], and as of now, there exists no Prolog compiler for the Nokia N900 architecture. It is unlikely that efforts will be undertaken to offer such a compiler any time soon, since even though SWI-Prolog is very efficient, the computational overhead created by Prolog does not make it a good language choice for mobile devices in general. TREF is the result of our previous research effort and we have dedicated Sect. 4 to briefly describe its functionality.

We have considered the fact that a cellular phone is not always guaranteed to have a decent network connection. Spots without a carrier signal, such as tunnels, elevators, etc. have to be taken into consideration. Therefore, the communication between the server and the client is done over asynchronous calls, to guarantee a service even in the case of an interruption of the network connection. Additionally, a database on the mobile device is kept to store user input, such as vocabulary lists or program preferences, and enable system use while the network and/or the server is not available. Outdated entries from this database are purged periodically, due to limited storage capabilities on the cell phone.

The overview of the server-client architecture is shown in Fig. 2. The *translation* module presents the contents graphically and provides input fields. The communication with the *TREF* module is initiated once a translation query is received. The *language learning module* offers a graphical learning program with various functionality, described in

Sect. 5. Data from the knowledge base on the server are used to construct a customized learning support, in terms of word/sentence suggestions and difficulty level. Both interfaces store essential data on the device. The language learning module accesses the *personal database* on the phone, which holds the necessary information to operate the framework without a network connection, though with limited capabilities. This personal database is synchronized with the *user-specific database* on the server, whenever possible. The user-specific database, as shown in Fig. 2, stores the user's history, such as previous query sentences. Words, compounds, and sentences stored in this database are assumed to be of interest to the student and are preferred in lecture suggestions. In future work, we plan to use ontologies in combination with the user's queries to find contextually fitting study suggestions.

The *didactic analysis module* makes decisions about the difficulty and the representation of the learning data by analyzing the user data from the user-specific database. The user can also provide information about his environment, e.g. his degree of attention, similar to [1]. We also plan to automatically infer as much information as possible by monitoring sensor data, to perform situation analysis by using GPS and noise level data.

In the future, we intend to integrate a voice recording module as well as a text capture module. The recording module will recognize words, spoken into the microphone of the cell phone, and translate and store them in the user's personal database. The text capture module will extract text from pictures, taken with the camera of the cell phone. The text in the picture will then be identified, translated, and stored in the personal database.

4. TRANSLATION ENHANCEMENT FRAMEWORK

Machine translation between natural languages, which differ significantly in surface characteristics (this applies to most European-Asian language pairs), is a difficult task, and existing approaches do not produce satisfying results. With our *TRanslation Enhancement Framework* (TREF) [30] we have succeeded to make a step forward in this area. Therefore, we apply these results in our *Ubiquitous TRanslatiOn and Language Learning* environment (UTROLL).

The underlying assumption of TREF is that there is a significant overlap between the **structure** of a sentence and its **meaning**. Based on this assumption TREF uses a sequence alignment algorithm by [11], taken from the field of bioinformatics, to make decisions about the restructuring of a translated sentence. As a translation basis we take the output of the statistical machine translation system *Moses* [9]. The overview of the dataflow in TREF is shown in Fig. 3. An input sentence is sent to the *part-of-speech* (PoS) tagger *MontyLingua* [13] for English, and ChaSen [15] for Japanese. After each sentence token is assigned a PoS-tag, the sentence and its tags are compared with sentences from a preformatted corpus. For this purpose, we have modified and enriched the *Jenaad* Corpus [23], a bilingual data collection consisting of 150,000 sentences, taken from news articles. We have removed as much noise as possible from the data, assigned PoS-tags to each sentence token and stored the information in an SQL database. We have created different formats of the bilingual data, one with a complete set of PoS information and others with reduced and optimized tag sets to provide quick access and efficient processing. Additional representations and tag sets can be added easily to satisfy different needs in future work. We have applied relational sequence alignment [11] to obtain clusters of structurally similar sentences, so that the comparison of the query sentence with the clusters yields several similar structures. At the same time, the query sentence is processed with Moses to obtain a preliminary translation. This translation is then used to fill the template of the structures, which had been found to be similar in terms of PoS-tags. This way, a certain number of translation candidates is produced. The parameters of the similarity measure can be adjusted to fine-tune the result, depending on the text type and text domain. Allowing low threshold values for similarity, a higher number of candidates can be produced, whereas a higher threshold value reduces the number of candidates.

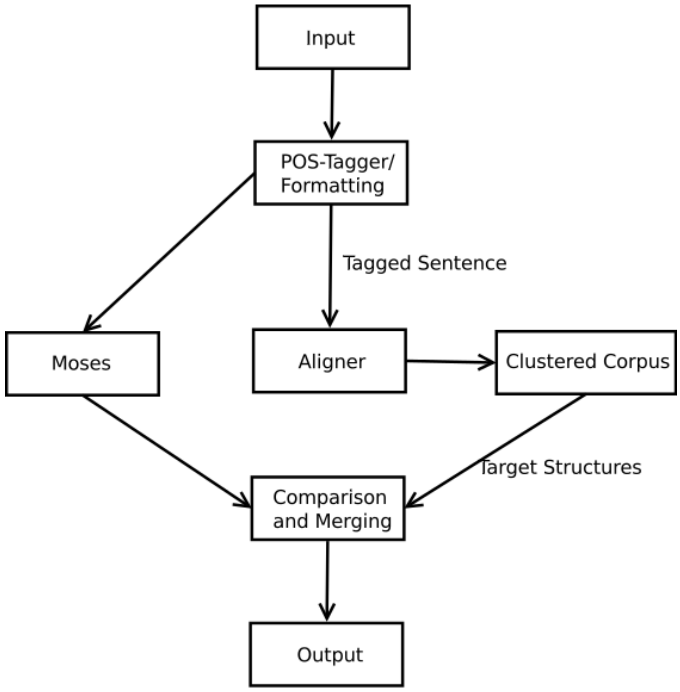

Figure 3: Overview of dataflow

The example in Fig. 4 illustrates how the sentence "My name is Yamada" is processed to present the user with the output, i.e. the translation candidates. The Japanese words in this schematic representation are written in Roman transcription. The acronyms represent the PoS-tags from MontyLingua and ChaSen, e.g. `NNP` (proper noun singular) or `VBZ` (verb, 3rd person singular present). First, the PoS-tagging of the input sentence is performed, in this case: `My/POP` (personal pronoun), `name/NN` (noun), `is/VBZ` (verb, 3rd person singular present), `Yamada/NNP` (proper noun). The alignment detects sentences in the database which are similar in terms of words and PoS-tags. In this step, different weights can be assigned to word or tag matches, so the output can vary. We still experiment with different parameter settings on large input data, to improve the alignment results. All structures of those similar sentences are considered structure candidates for the final translation. The two groups of sentences in the figure represent the concept of *structure-to-*

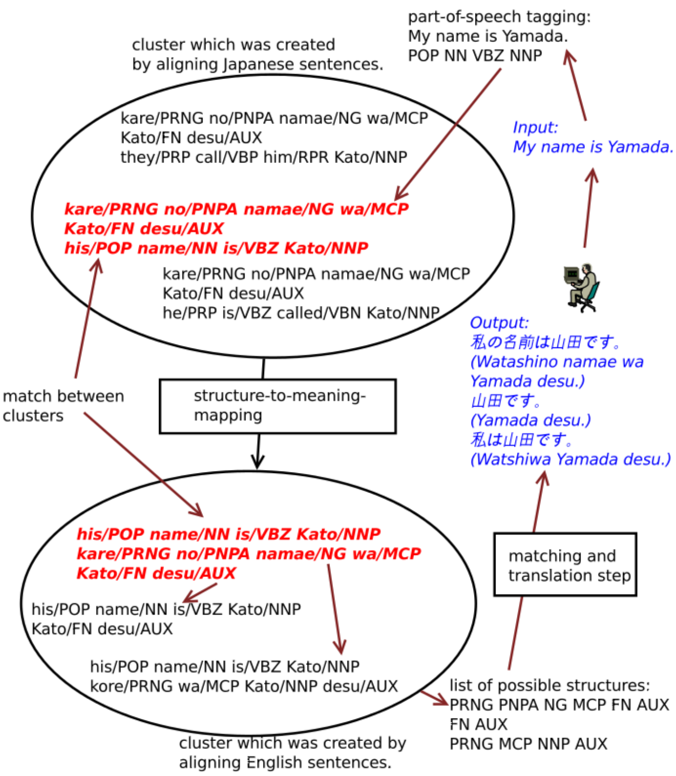

Figure 4: Translation via clustering [30]

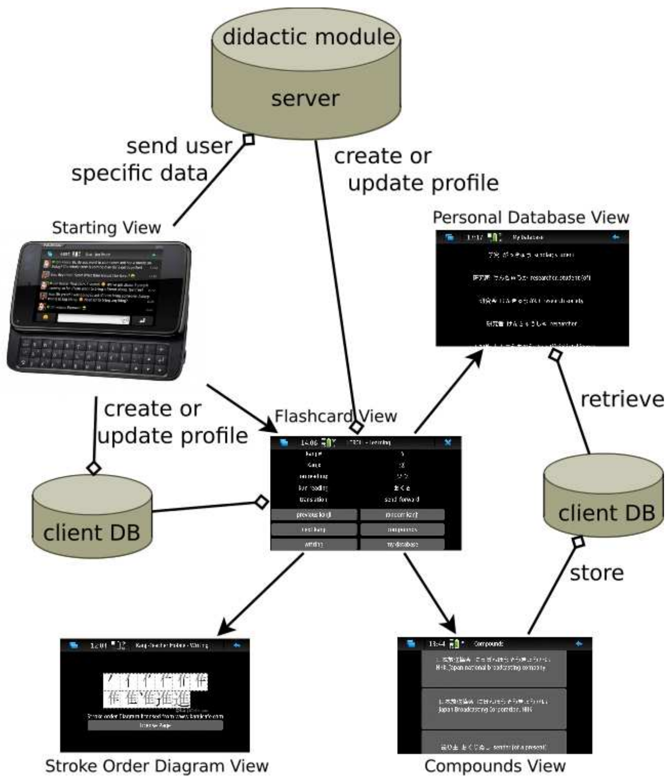

Figure 5: UTROLL – Language Learning Task

meaning-mapping as mentioned before. The candidates obtained from this mapping are sent to the *matching and translation* step to improve the translation result from Moses.

Once the translation is finished, the target sentence is sent back and displayed on the client. As a useful feature in terms of language learning, the user can choose to display the PoS-tagged sentence from the translation process to obtain an insight about the linguistic characteristics. In the future we plan to integrate a search for similar sentences based on the aligning algorithm to find similar sentences for study suggestions.

The interface of TREF for the communication with the client (described in Sect. 3) is the Django Web framework (`www.djangoproject.com`). In our previous work we have used Django to build a Web site interface. This site provides the access to the TREF module and includes the translation functionality, PoS-tag output, random sentence output from the Jenaad corpus as well as legends for the PoS-tags of MontyLingua and ChaSen. The translation of the original Japanese ChaSen tags into English was part of our previous work and is, to the best of our knowledge, the only English ChaSen PoS-tag legend available. The entire content is accessible through the Web interface at `https://wloka.dac.univie.ac.at/project/`.

5. SHOWCASE

In this section we present the currently implemented components of our *Ubiquitous TRanslatiOn and Language Learning* (UTROLL) environment. At the time of writing, we have implemented a Japanese-English translation module (see below), and a *kanji* learning module as part of the language learning task of UTROLL. Kanjis are Japanese characters, adopted from China roughly 1,500 years ago. Due to their origin, these characters have at least a traditional Chinese and a Japanese pronunciation, and often several meanings, depending on the context or the combination with other characters. Because of this complexity, we have decided to start our language learning task with a kanji learning module.

The workflow of the kanji learning module and its GUI are demonstrated in Fig. 5. In the *Starting View* the user has the choice of starting the *Translation Task* or the *Language Learning Task*. Upon initiating the Language Learning Task, there is an attempt to connect to the server. If the connection is successful, the user-specific data is transmitted to the server, where the didactic profile is generated, according to the data in the user-specific database. The result, which is sent back to the phone, determines the difficulty level of the learning material. In order to maintain a challenging learning experience with growing skill, the didactic analysis is performed periodically.

In the kanji learning application, users can review flashcards with various information about a kanji, which they can choose to be displayed randomly or in a sequence suggested by the didactic profile. We call this the *Flashcard View* (Fig. 6). From here, the student can switch to the *Stroke Order Diagram View* (Fig. 7), where the exact way of drawing the kanji is presented with a sequence of images. We strictly follow the *cognitive load theory* mentioned in Sect. 2, by separating the information into different views. The *Compound View* (Fig. 8) is a collection of compound terms, extracted from *JMdict* [3], which include the currently selected kanji. Users can store any of those com-

Figure 6: Flashcard View

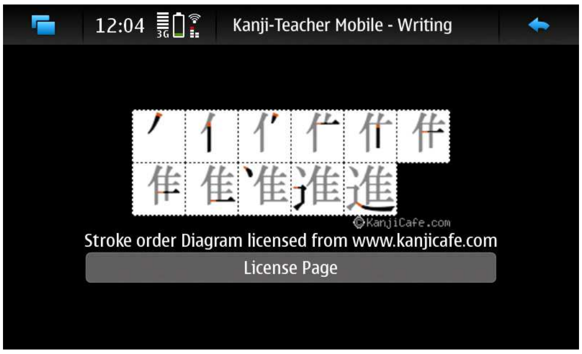

Figure 7: Stroke Order Diagram View

pounds in a personal database. These stored compounds can later be reviewed in the *Personal Database View* (Fig. 9), which is accessible from the Flashcard View. The stored entries are considered in the process of creating or updating a didactic profile, in a way that the future suggestions will emphasize on similar terms or the domain of the terms.

Selecting the *Translation Task* from the Starting View initiates a workflow as shown in Fig. 10. Upon initiating the task, a connection to the server is attempted. In contrast to the language learning task, a connection to the server is vital. In the first view (Fig. 11), the user has the choice of the input language. An input box (Fig. 12), takes the query sentence and sends it to the TREF module, which is located at the server and described in Sect. 4. The Japanese input can be done via SCIM (Smart Common Input Method)

Figure 8: Compounds View

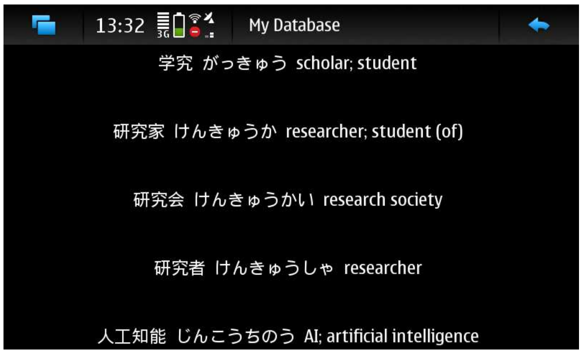

Figure 9: Personal Database View

usually available on every Linux-based system. In the next view (Fig. 13), the translation result is displayed and the query sentence is stored on the server for later use by the didactic module. Additionally, the user has the choice of viewing further language details (Fig. 14). At the time of writing, we display various PoS-tag information. In the future we plan to integrate a more detailed linguistic analysis, e.g. dependency trees produced by *CaboCha* [12].

Even though the current programming libraries available for the Maemo5 platform are fairly intuitive and easy to program, there is still a lack of certain design capabilities, so that there has been a significant discrepancy between our initial design and the final realization. Unsightly gaps between the lines or between buttons are almost impossible to avoid using the current version of the SDK.

The final implementation of the application on the Maemo platform is available for download at: `http://maemo.org/packages/view/ktmobile/`.

6. CONCLUSION

In this paper we have described the design and a partial implementation of a ubiquitous translation and language learning framework, in particular for Japanese-English, on the Nokia N900, a new cellular phone with internet tablet capabilities. We have presented our implementation of a translation and language learning environment built as a client-server system, which consists of a translation and a language learning task. For the language learning task we have built a kanji learning application including a didactic module, and incorporated the findings of current cognitive load theory research for an improved learning experience. For the translation task we used TREF, a translation enhancement framework, which we had implemented previously. By integrating TREF into this research work, we have shown how a client/server configuration can be realized to offer the entire translation service on the mobile device. At the time of writing we have tested the translation module on the server as well as parts of the client-side application. We plan to address the issue of security and the protection of personal data while sending the information over the network.

In future work, we intend to integrate more pervasive elements into our framework. We plan to use the sensors and the software to monitor the user as well as the situation of the user, to further refine the learning experience. In particular, we want to focus on audio input, GPS localization, and

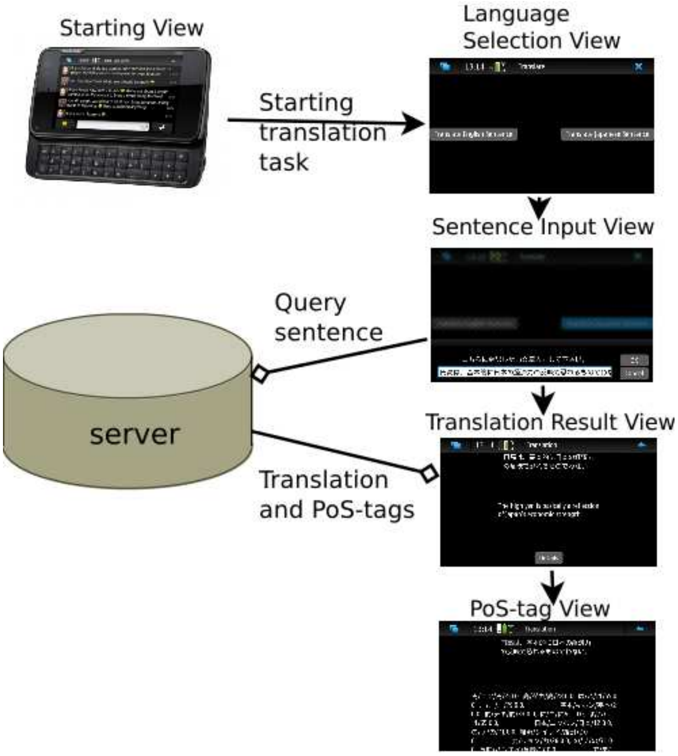

Figure 10: UTROLL – Translation Task

Figure 11: Language Selection View

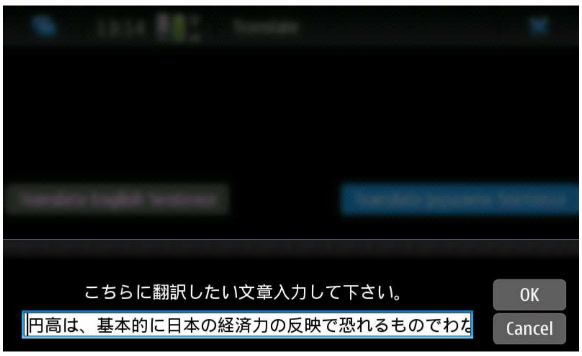

Figure 12: Sentence Input View

Figure 13: Translation Result View

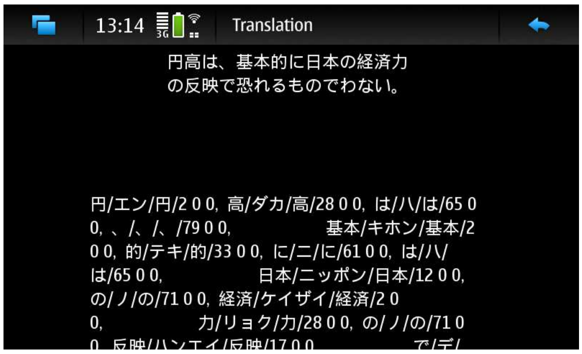

Figure 14: PoS-tag View

user query statistics to create a detailed user profile. This will facilitate a specific fine-tuning of the learning environment, considering guidelines for computer enhanced learning such as the cognitive load theory, and the GUI design, as mentioned in Sect. 2. In order to reduce the cognitive load even further, the learning material will be displayed in different forms. We have recently found that one ideal candidate for that purpose is the Japanese *WordNet* [2], as it also offers a growing collection of graphical illustrations of concepts.

We plan to extend the translation component by adding a functionality which will extract and translate captions from photos, taken with the built-in camera, as well as transcribe and translate speech recorded with the microphone.

As mentioned before, the already implemented kanji learning environment is available on the Web to receive feedback from the Nokia N900 community. In addition, we will make our framework available to students at our Language Department, once it is in a workable state. We will evaluate the framework with a sufficient number of users and a control group along the dimensions engagement, effectiveness, and viability [17], as well as analyze the usability on other prevalent mobile platforms, such as iPhone, Android, Maemo5's successor MeeGo, etc.

7. REFERENCES

[1] B. Bomsdorf. Adaptation of learning spaces: Supporting ubiquitous learning in higher distance education. In N. Davies, T. Kirste, and H. Schumann, editors, *Mobile Computing and Ambient Intelligence: The Challenge of Multimedia*, number 05181 in

Dagstuhl Seminar, Dagstuhl, Germany, 2005.

[2] F. Bond et al. Enhancing the Japanese WordNet. In *Proceedings of the 7th Workshop on Asian Language Resources, in conjunction with ACL-IJCNLP*, 2009.

[3] J. Breen. JMdict: A Japanese-Multilingual dictionary. In *COLING Multilingual Linguistic Resources Workshop*, 2004.

[4] C. Calvary et al. Plasticity of user interfaces: A revised reference framework. In *Proceedings of TAMODIA 2002*, 2002.

[5] Y. Chen et al. A mobile scaffolding-aid-base bird-watching learning system. In *Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Eduction*, pages 15–22. IEEE Computer Society Press, 2002.

[6] Y. Cui and S. Bull. Context and learner modelling for the mobile foreign language learner. *System*, 33(2):353–367, 2005.

[7] L. H. Gan et al. Language learning outside the classroom using handhelds with knowledge management. In *Proceedings of the 2007 Conference on Supporting Learning Flow through Integrative Technologies*, pages 361–368, Amsterdam, The Netherlands, 2007. IOS Press.

[8] C. Goutte et al., editors. *Learning Machine Translation*. MIT Press, Cambridge, Massachusetts, 2009.

[9] H. Hoang et al. Moses: Open source toolkit for statistical machine translation. pages 177–180, 2007.

[10] V. Jones and J. H. Jo. Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology. In *Proceedings of the 21st ASCILITE Conference*, 2004.

[11] A. Karwath and K. Kersting. Relational sequence alignments and logos. pages 290–304, 2007.

[12] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[13] H. Liu and P. Singh. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[14] K. Lyytinen and Y. Yoo. Issues and challenges in ubiquitous computing. *Commun. ACM*, 45(12):62–65, 2002.

[15] Y. Matsumoto, A. Kitauchi, T. Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. *Japanese Morphological Analysis System ChaSen version 2.2.1*, 2000.

[16] N. Nagata. Banzai: Computer assisted sentence production practice with intelligent feedback. In *Proceedings of the Third International Conference on Computer Assisted Systems for Teaching and Learning/Japanese (CASTEL/J)*, 2002.

[17] D. A. Norman and J. C. Spohrer. Learner-centered education. *Commun. ACM*, 39(4):24–27, 1996.

[18] H. Ogata. Computer supported ubiquitous learning: Augmenting learning experiences in the real world. In *IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education*, pages 3–10, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[19] H. Ogata et al. Computer supported ubiquitous learning environment for Japanese mimicry and onomatopoeia with sensors. In *Proceedings of the 2007 Conference on Supporting Learning Flow through Integrative Technologies*, pages 463–470, Amsterdam, The Netherlands, 2007. IOS Press.

[20] H. Ogata and Y. Yano. Context-aware support for computer-supported ubiquitous learning. In *Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04)*, page 27, Washington, DC, USA, 2004. IEEE Computer Society.

[21] M. A. Rau, V. Aleven, and N. Rummel. Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education*, pages 441–448, Amsterdam, The Netherlands, 2009. IOS Press.

[22] J. Sweller, J. van Merrienboer, and F. Paas. Cognitive architecture and instructional design. *Educational Psychology Review*, pages 251–296, 1998.

[23] M. Utiyama and H. Isahara. Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 72–79, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[24] M. Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.

[25] J. Wielemaker. An overview of the SWI-Prolog programming environment. In F. Mesnard and A. Serebenik, editors, *Proceedings of the 13th International Workshop on Logic Programming Environments*, pages 1–16, Heverlee, Belgium, 2003. Katholieke Universiteit Leuven.

[26] Y. Wilks. *Machine Translation: Its Scope and Limits*. Springer-Verlag, 2008.

[27] W. Winiwarter. WETCAT – Web-Enabled Translation using Corpus-based Acquisition of Transfer rules. In *Proceedings of the Third IEEE International Conference on Innovations in Information Technology*, Dubai, United Arab Emirates, 2006.

[28] W. Winiwarter. JETCAT - Japanese-English Translation using Corpus-based Acquisition of Transfer rules. *JCP*, 2(9):27–36, 2007.

[29] W. Winiwarter. WILLIE – a Web Interface for a Language Learning and Instruction Environment. In *Proceedings of the 6th International Conference on Web-based Learning*, Edinburgh, United Kingdom, 2008. Springer-Verlag.

[30] B. Wloka. Enhancing Japanese-English machine translation – a hybrid approach. Master's thesis, University of Freiburg, 2009.

[31] C. Yin, H.Ogata, and Y. Yano. JAPELAS: Supporting Japanese polite expressions learning using PDA(s) towards ubiquitous learning. *International Journal of Information and Systems in Education*, 3(1):33–39, 2005.