Internationalisation and localisation of spoken dialogue systems

Niklas Laxström, Graham Wilcock and Kristiina Jokinen

Abstract In contemporary software development, localisation is a straightforward process–assuming internationalisation has been considered during development. The localisation of spoken dialogue systems is less mature, possibly because they differ from common software in that interaction with them is situated and uses multiple modalities. We claim that it is possible to apply software internationalisation practices to spoken dialogue systems and that it helps the rapid localisation of such systems to new languages. We internationalised and localised the WikiTalk spoken dialogue system. During the process we identified needs relevant to spoken dialogue systems that will benefit from further research and engineering efforts.

1 Introduction

A spoken dialogue system is a software application that is able to converse in speech. It accepts input from the speech recogniser, interacts with external knowledge sources, and produces messages as output to the user [14]. An important component is the dialogue manager which enables interaction with the user, and generally controls the dialogue flow.

Speech interfaces are useful in situations where the hands are not free. They provide access to digital databases and allow the user to search for information using natural language. Since the 1990s, a large number of speech-based systems have been developed that support mobile services, car navigation, and various phone-based voice applications. Personal assistants with speech interfaces on smart phones are becoming commonplace, for example Siri and Cortana. In addition, spoken dialogue systems are now essential for humanoid robots which are expected to move rapidly into everyday use.

University of Helsinki, Helsinki, Finland e-mail: `first.last@helsinki.fi`

As smart phones propagate around the world, often being people's only way to access the Internet, localisation of personal assistants is as important as for other software. A particularly challenging area is robotics, where spoken dialogue systems need to maintain a two-way conversation by listening as well as talking with the situated agent, and focus on the physical environment as well.

The importance of localisation is widely acknowledged in the software industry: localisation is considered to be necessary for the software developers to remain competitive [26, 21]. Localisation removes the language barrier, making the software usable for more people. It also allows people to work using their most fluent language.

Localisation helps preserve non-dominant languages by encouraging their use and make them visible in the digital world. Localisation can also serve to revitalise endangered languages by motivating their speakers to use them. For instance, [17] points out that to survive digitalisation, a language has to have a digitally performed function, i.e. it has to produce new, publicly available digital material.

The DigiSami project[1] focuses on the viability of the the North Sami language (one of the small Finno-Ugric languages spoken in the Northern Europe) by supporting digital content generation and experimenting with language technology applications that can strengthen the user's interest in using the language in various interactive contexts. For the latter purpose it uses WikiTalk[31].

We investigated whether common software internationalisation and localisation methods are applicable to spoken dialogue systems. Our hypotheses are:

1. existing software internationalisation practices can be applied to spoken dialogue systems;
2. once internationalisation is in place, localisation of a spoken dialogue system to a new language is rapid.

To test the hypotheses we performed a practical case study using the WikiTalk spoken dialogue system described in Section 3. WikiTalk was initially developed without consideration for internationalisation and localisation. For this study we first created an internationalised version of WikiTalk to test the first hypothesis.

We also localised WikiTalk to English and Finnish. We then involved other people to localise WikiTalk to Japanese in order to test the second hypothesis. Although in this study we used WikiTalk with a Nao robot, most of the observations apply to spoken dialogue systems in general.

The paper is structured as follows. Section 2 summarises previous work on the theory and methods of internationalisation and localisation of spoken dialogue systems and of software. Section 3 describes WikiTalk. Section 4 contains the results of the study and Section 5 discusses the limitations of our internationalisation. Section 5.2 addresses issues specifically related to the Nao robot and multimodality. Finally, we discuss the results and conclude with recommendations for future improvements to the internationalisation of spoken dialogue systems.

[1] http://www.helsinki.fi/digisami/

2 Previous work and methods

Software internationalisation and localisation has a long history [8]. We can therefore adopt long-standing software internationalisation practices and apply them to spoken dialogue system software.

While there is much standardisation in software internationalisation, there is great variance in its methods and quality. Hence there are no unified practices [6]. However, we consider libraries such as GNU Gettext [1], in use for two decades, to encapsulate a stable core of practices. The Gettext ecosystem is well documented [27]. In addition we rely on our experience working with internationalisation and localisation of many open source projects which use various libraries.

2.1 Software internationalisation and localisation

Internationalisation is the process of designing software so that it can be adapted to different languages or regions. Localisation is the process of adapting the software to a specific locale. World Wide Web Consortium (W3C) provides a concise summary of the concepts [34].

Internationalisation and localisation are closely related. Internationalisation is done by the software developer to make localisation possible and easy. Localisation is usually done by an expert in a language, often known as a translator even though the localisation work is not only translation.

The goal of internationalisation is to facilitate the best software localisation while minimising changes required to the software source code. Poor implementation of internationalisation restricts the quality of localisation. For example, if the software places restrictions on the word order of an interface string, it is impossible to produce good localisation in languages where the natural word order differs. This issue is solved in Example 1, which shows a string in English (a) translated into German (b). A good German translation would be impossible if the translator was only able to translate the word *Show*. Because in the example we have a placeholder (`$1`) that can be moved, a good translation is possible.

(1) a. Show $1?
 b. $1 anzeigen?

It is common to implement internationalisation with programming libraries. Such libraries are beneficial in many ways: they reduce implementation costs and increase interoperability with various tools used by the translators. In practice, however, there are many different internationalisation libraries that are not completely compatible with each other. Interoperability between libraries and translation tools is most often achieved only through the file format of the interface strings. For example, web-based translation platforms such as Pootle or translatewiki.net are able to provide translators with an easy interface and various translation aids (e.g. translation memory), as long as they support the given file format.

2.2 Dialogue system localisation

A common way to localise spoken dialogue systems is to have a general dialogue model with well-defined dialogue acts which are considered universally valid, and then realise the communicative intentions with respect to different languages. The system's presentation language is localised with respect to the user's actual language. The dialogue model approach has often been considered one of the main benefits for building separate and hierarchical modules of a more general model of interaction, while leaving realisation of the dialogue acts to separate language-specific components, so that language generation is a task for localisation [10].

There has been work on standardising dialogue acts and interaction management, and this has resulted in ISO standards [3] which aim at a general and scalable set of dialogue acts which could be used amongst different tasks, activities and languages. However, very few dialogue systems are actually built on these theoretically and practically well-grounded acts. The reason may be that for a simple voice interface such a set of dialogue acts is considered too elaborate, or maybe the acts are considered too theoretical for an engineering task of building practical interfaces.

Dialogue strategies are usually designed for a user who is a generic mean of the possible users. Adaptation to different users is done via user modelling which takes the various users' individual preferences into account: this can range from a list of preferences related to language, speed, etc. to a more elaborated modelling of the user's likes and habits as in recommendation systems. The system can use the user model to provide suitably tailored help [11] or proactively take the user's needs into account [23].

Another problem in localising spoken dialogue systems is that interaction management also requires localisation. It is much more difficult to standardise spoken interaction features such as turn-taking practices, politeness codes, and feedback realisation with the help of standard interaction management protocols. The common patterns of interaction with system initiative dialogues ("Please give the name of the arrival city") and clarification question ("Did I hear you say...") are common practices, but render the systems rather clumsy and unnatural. Natural interaction features vary depending on the culture and language, and also vary for each individual, so various large dialogue modelling studies are conducted, and also intercultural studies are necessary to model the differences. It may not be easy or even possible to have a standard set of interaction patterns which can then be localised for interaction management in various tasks, activities, contexts and cultures.

3 WikiTalk

WikiTalk [31] is a spoken information access system that gets its information from Wikipedia. The system talks about a given topic using sentences extracted from the Wikipedia article about the topic. The sentences are processed to make them more suitable for spoken dialogue, mainly by removing large amounts of textual details (parenthetical information, footnotes, infoboxes and so on). The hyperlinks in the Wikipedia text are also extracted and are used to predict the user's utterances in order to support smooth shifts to related topics. Example dialogues with WikiTalk are given by [15, 31].

WikiTalk is bound by the limitations of the automatic speech recognition (ASR) and text-to-speech (TTS) systems available on the robot. The Nao robot currently supports ASR and TTS for 19 different languages (which must be purchased separately, apart from English and one other language). An important limitation is that ASR on Nao can process only one language at a time. This makes it difficult to handle language switching in multilingual situations and rules out multilingual conversations.

Nao does not provide language identification of speech. Such technology exists [19] but is not readily available. Language identification of text has become a common tool in translation agencies and machine translation services. In the case of Nao, implementing spoken language identification is expected to be complicated due to limited processing power on the robot. Spoken language systems would benefit from such a technology: in WikiTalk it would allow more fluent language selection. Our language selection mechanisms are described in Sections 4.1 and 4.3.

In WikiTalk the interface language, the language the user uses to communicate with the application, is linked to the Wikipedia in the same language. This means that to have WikiTalk in a particular language, we must have all of the following: localisation of WikiTalk in that language; ASR and TTS for that language; and a Wikipedia for that language.

4 WikiTalk internationalisation

We start with the results regarding the first hypothesis. Section 4.4 discusses results regarding the second hypothesis of rapid localisation.

Generally the software internationalisation process goes as follows: First choose an internationalisation library. Using the library, make the interface strings translatable; adapt the code to handle other internationalisation features (character encoding, number formatting, calendar systems, etc.) depending on the needs of the application; and implement language selection.

We implemented the internationalisation step by step: after each feature we decided what to add next by seeing what would make the biggest difference in the quality of the localisation. The following sections contain more details of each step.

4.1 String extraction, variables and language selection

First, we had to identify the software interface strings. In WikiTalk, those are the available user commands and the interaction management utterances by the robot.

For example, when the user interrupts the speaking robot, the robot says in English *oh sorry*, in Finnish *anteeksi* and in Japanese *sumimasen*. This process, known as string extraction, is a typical step when internationalisation is done after development [4, 30]. The aim is to move the strings to be translated into a file separate from the application code, by replacing the string in the code with a function call. This function needs to know which translation it should fetch. Therefore, an identifier was assigned to each extracted string. We stored the map of identifiers and string contents in a JSON file format, one file per language.

Further work was necessary on these strings as some translated utterances were awkward or even ungrammatical due to differences in word order. We solved this issue by adopting variables, which act as movable placeholders in the translatable strings (see Example 1 in Section 2.1).

Once localised, every software needs a way to select what locale to use. Given the limitations in language identification (see Section 3), we added a configuration option that sets the default language to use. When WikiTalk starts up, it announces the availability of other languages. The robot says *say Finnish if you want to use Finnish* in Finnish, and continues similarly for other available languages. This approach does not scale above a few languages: it would take a considerable amount of time to list even ten languages, not to mention hundreds of languages supported by some software applications.

4.2 Wikipedia content processing

A common practice in software internationalisation is to make the code generic enough to work in any language. Programming interfaces are used to abstract the differences behind a common interface.

Natural language processing is one WikiTalk component for which this principle is important. In order to avoid the difficulties of open-vocabulary speech recognition, WikiTalk predicts the user's next utterance and only listens for a specified vocabulary of words and phrases. The recognition vocabulary is dynamically generated according to the dialogue context.

Wikipedia content processing in WikiTalk is summarised by [18]. We used the Python Beautiful Soup library to extract the plain text of Wikipedia articles and the Natural Language Toolkit library to tokenize the text into sentences. Information about links is used both to synchronise gestures (see Section 5.2) and to define the keywords the robot listens for to make topic shifts.

4.3 Finnish WikiTalk

WikiTalk listens for language autonyms like "suomi" (Finnish) using English language ASR. We used this as an additional mechanism for language selection. This solution only works in this particular context. As an experiment, we tried to decouple the *interface language* from the *Wikipedia language*. We accessed Swedish Wikipedia using Finnish WikiTalk and found the quality of Swedish using Finnish TTS understandable but unnatural. The experience was comparable to using machine translation when proper translation is not available.

We did not use an internationalisation library to support Finnish localisation in WikiTalk as it would be faster and easier to adapt our own implementation to any special needs. Many internationalisation libraries are available for Python [27], the programming language of WikiTalk: a comparison shows that our development reimplemented many of their common features, supporting our first hypothesis.

4.4 Japanese WikiTalk

Following the internationalisation of WikiTalk and the creation of English and Finnish localisations, the next step was to make a Japanese localisation to test the second hypothesis. While English and Finnish use variants of the Latin writing system, Japanese uses a quite different writing system and is a good example for testing whether the language support is generic enough. The Japanese localisation is described by [24].

The differences between the English and Finnish alphabetic characters and the Japanese kanji, hiragana and katakana characters were handled by using Unicode. The main problem was caused by the fact that both English and Finnish separate words by spaces unlike Japanese. For the Japanese localisation the code was rewritten so that it does not assume spaces between words.

The English WikiTalk uses the *Did you know?* section from the main page of English Wikipedia in order to suggest new topics that might be interesting. The topics in this section are new every day. Japanese Wikipedia does not have a *Did you know?* section, so this method could not be used unchanged. However, Japanese Wikipedia does have a list of new articles on the main page, and this list was used to suggest new topics in a similar way.

For language-switching, we found that the workaround mentioned in Section 4.3 also worked for Japanese. English ASR recognises *Nihongo* as a request to switch to Japanese, and Japanese ASR recognises *Ingirishu* (English) as a request to switch to English. Language-switching between English and Japanese with multilingual WikiTalk was demonstrated at SIGDIAL 2015 [32].

5 Limitations on internationalisation

The previous section describes an implementation of internationalisation that makes usable localisations possible. Existing and future localisations, however, could benefit from additional internationalisation support. In this section we describe additional requirements which are not universally adopted in standard software internationalisation and hence were not provided in our study. Solving these issues would be the next steps for increasing the quality of internationalisation support for spoken dialogue systems such as WikiTalk.

5.1 Grammatical features in text interfaces

Often, the translated string needs to be different depending on the actual value that will replace a placeholder. This is still an on-going engineering problem in software development. For number agreement this is a solved problem, however, as most internationalisation libraries let translators specify different versions depending on a numerical value. The rules for this are standardised and available in the Unicode Common Locale Data Repository [29]. Example 2 demonstrates one such system. An interface string (a) shows the syntax which translators use to specify different versions: with value 1 the chosen version is (b) and with 5 (c).

(2) a. $1 {{PLURAL:$1|day|days}} ago
b. 1 day ago
c. 5 days ago

Besides grammatical number, there are many other types of agreements that cannot be handled by common internationalisation libraries. In some software, such as those used by Wikipedia and Facebook, it is possible to alter the string based on the gender of the placeholders.[2] In fact there was very little need for grammatical features for WikiTalk.

5.2 Issues in multimodal interfaces

The approach to internationalisation and localisation for WikiTalk treats the system like a normal software project, but the speech and multimodal interfaces raise additional issues. ASR and TTS systems are clearly less mature in terms of internationalisation than text-based interfaces. While in most parts of the world people can use computers where all the necessary fonts and input methods are available for text, the same is not true for speech. ASR and TTS systems in many languages are either missing or low quality.

The technology for multimodal interaction (including face tracking, nodding and gesturing) is even less mature than for speech. Existing internationalisation libraries provide no support for issues related to multimodal interaction, although research on

[2] Support for gender is complicated given that natural gender and grammatical gender might be different in some languages. It becomes even more complex when a user of the software uses a different language than the user who is being referred to.

multimodal interaction clearly shows that there are significant differences between languages for all these modalities.

Some of these differences are well-known, for example gesturing with hands and arms varies widely between cultures. The Nao robot provides a large collection of predefined gestures, but they are not language-dependent. These predefined gestures are intended to express emotions such as surprise and embarrassment, but the way these emotions are actually expressed in different communities varies greatly and in subtle ways. In addition to the predefined gestures, applications can develop their own sets of gestures. WikiTalk uses its own gestures which are classified into gesture families according to their communicative functions, as discussed in [20].

Other differences in multimodal behaviours are less well-known. For example, head nodding can vary even between language communities that are geographically close to each other. The NOMCO multimodal Nordic video corpus [25] shows significant differences in head nodding between Danes, Swedes and Finns. Danes tend to make multiple rapid down-nods, Swedes tend to make up-nods relatively more often, and Finns tend to make a single extremely slow down-nod [22, 28].

Face tracking also needs to be localised. In some cultures it is normal to look straight at the eyes of the dialogue partner, in other cultures it is normal to avoid direct eye-contact, and in many cultures it depends on parameters of the situation which can be quite complex (gender, age, social status, and so on). The Nao robot performs face tracking rather well, but the only options are to switch face tracking on or switch it off, and when it is on the robot looks straight at the detected face. Any parametrisation of face tracking in order to support localisation would require replacing the standard face tracking module with custom software. WikiTalk currently includes only the option of running with face tracking on or off.

The proximity between the robot and the user is another factor in multimodal interaction. Unlike a computer, a humanoid robot can walk and can move closer to the user or away from the user. Initial measurements of users' preferred distance from the robot are reported by [9], but it is well-known that preferred proximity between human speakers varies between cultures. This suggests that proximity is another multimodal feature of human-robot interaction that requires localisation.

Apart from localisation of individual modalities, there is the further problem of integration across modalities. In WikiTalk, one of the most difficult challenges is the synchronisation of speech and gestures. The utterances spoken by the robot include some words and phrases that should be emphasised by specific gestures. In WikiTalk, *NewInfos* are pieces of new information that typically correspond to hyperlinks in the Wikipedia text, and these NewInfos should be emphasised by a *beat* gesture [5]. Like most gestures, beat gestures have a preparation phase before the main beat phase. Synchronisation of the beat gesture with the NewInfo therefore requires estimating both the time required for the preparation phase of the gesture and the time required for uttering the words that precede the NewInfo word or phrase. The problem for localisation is that the number of words preceding the NewInfo may vary greatly between languages, and it remains to be confirmed whether the methods for synchronising speech and gestures described by [20] can be made to work successfully across different languages.

6 Conclusions and future work

While this study is based on only one spoken dialogue system, the results are promising and support our hypotheses. We found that many existing software internationalisation practices and libraries are applicable for spoken dialogue systems. We also found interesting special internationalisation needs that are not present in common software applications.

The Japanese WikiTalk localisation was completed rapidly, proving that our internationalisation implementation for the WikiTalk spoken dialogue system enables rapid localisation. We have not evaluated the localised versions due to our assumption that localisation is not the bottleneck in naturalness of the conversations. The original WikiTalk was evaluated as described in [7].

We plan to add further locales. For example in the DigiSami project [12, 16] we plan to create a version of WikiTalk in Sami language (named SamiTalk). This work is described further in [33, 13].

This study has revealed special needs in internationalisation of spoken dialogue systems which are not yet adequately addressed in the common internationalisation libraries and frameworks. These special needs include identification of spoken language and many issues in multimodal interaction.

Computer gaming is probably the closest industry where related issues are found, together with localisation challenges presented by puns and sounds [2]. Future work should investigate whether spoken dialogue systems can borrow some ideas from lessons learnt by gaming. It might also be worth investigating the solutions for subtitle localisation which deal with multiple modalities of video, speech and text.

To understand how to best support these needs, further research and practical work should be conducted. For example, as it is currently impractical to provide a robot for each translator, further work could integrate robot emulators in tools used by translators. The aim is that translators would (in addition to translating the strings) be able to create appropriate localised gestures and tweak timings so that all modalities are properly synchronised.

Even new products targeting mass markets do not always support localisation from the beginning. This kind of development makes well-resourced languages even more privileged and increases the risks that less-resourced languages will be absent from new technologies. We recommend the developers of spoken dialogue systems to consider internationalisation from the beginning of the development and to take advantage of existing libraries, in order to avoid wasted effort later. For developers of internationalisation frameworks, we recommend providing support for localisation of multimodal interaction. Translation of strings is not sufficient to ensure a good experience for users of spoken dialogue systems. Developers of devices used for spoken dialogue systems should provide spoken language identification.

References

1. GNU Gettext. URL https://www.gnu.org/software/gettext/. Accessed: 2015-09-18
2. Bernal-Merino, M.: Translation and Localisation in Video Games: Making Entertainment Software Global. Routledge Advances in Translation Studies. Taylor & Francis (2015)
3. Bunt, H., Alexandersson, J., Choe, J.W., Fang, A.C., Hasida, K., Petukhova, V., Popescu-Belis, A., Traum, D.R.: ISO 24617-2: A semantically-based standard for dialogue annotation. In: Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 430–437. Istanbul (2012)
4. Cardenosa, J., Gallardo, C., Martin, A.: Internationalization and localization after system development: a practical case. In: 4th International Conference on Information Research, Applications and Education, vol. 1, pp. 207–214 (2007)
5. Cassell, J.: Embodied conversation: integrating face and gesture into automatic spoken dialogue systems (1999)
6. Cos, J., Toval, R., Toval, A., Fernandez-Aleman, J., Carrillo-de Gea, J., Nicolas, J.: Internationalization requirements for e-learning audit purposes. In: Global Engineering Education Conference (EDUCON), 2012 IEEE, pp. 1–6 (2012). DOI 10.1109/EDUCON.2012.6201027
7. Csapo, A., Gilmartin, E., Grizou, J., Han, J., Meena, R., Anastasiou, D., Jokinen, K., Wilcock, G.: Multimodal conversational interaction with a humanoid robot. In: Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012), pp. 667–672. Kosice (2012)
8. Esselink, B.: The evolution of localization. The Guide from Multilingual Computing & Technology: Localization **14**(5), 4–7 (2003)
9. Han, J., Campbell, N., Jokinen, K., Wilcock, G.: Investigating the use of non-verbal cues in human-robot interaction with a Nao robot. In: Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012), pp. 679–683. Kosice (2012)
10. Jokinen, K.: Three challenges for dialogue management: the constructive dialogue model approach. Teoksessa T. Haukioja (toim.) Papers from the 16th Scandinavian Conference of Linguistics pp. 221–234 (1996)
11. Jokinen, K.: Adaptation and user expertise modelling in AthosMail. Universal Access in the Information Society **4**(4), 374–392 (2006)
12. Jokinen, K.: Open-domain Interaction and Online Content in the Sami Language. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2014) (2014)
13. Jokinen, K., Hiovain, K., Laxström, N., Rauhala, I., Wilcock, G.: DigiSami and digital natives: Interaction technology for the North Sami language. In: Proceedings of Seventh International Workshop on Spoken Dialogue Systems (IWSDS 2016). Saariselkä (2016)
14. Jokinen, K., McTear, M.: Spoken dialogue systems. Synthesis Lectures on Human Language Technologies **2**(1), 1–151 (2009)
15. Jokinen, K., Wilcock, G.: Constructive interaction for talking about interesting topics. In: Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2012). Istanbul (2012)
16. Jokinen, K., Wilcock, G.: Community-Based Resource Building and Data Collection. In: The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14). St. Petersburg, Russia (2014)
17. Kornai, A.: Language death in the digital age. Language (1/8) (2012)
18. Laxström, N., Jokinen, K., Wilcock, G.: Situated interaction in a multilingual spoken information access framework. In: Proceedings of the Fifth International Workshop on Spoken Dialog Systems (IWSDS 2014), pp. 161–171. Napa, California (2014)
19. Li, H., Ma, B., Lee, K.A.: Spoken language recognition: from fundamentals to practice. Proceedings of the IEEE **101**(5), 1136–1159 (2013)
20. Meena, R., Jokinen, K., Wilcock, G.: Integration of gestures and speech in human-robot interaction. In: Proceedings of 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012), pp. 673–678. Kosice (2012)

21. Murtaza, M., Shwan, A.: Guidelines for Multilingual Software Development. Chalmers University of Technology, University of Gothenburg, Department of Computer Science and Engineering, Gothenburg, Sweden (2009)
22. Navarretta, C., Ahlsén, E., Allwood, J., Jokinen, K., Paggio, P.: Feedback in nordic first-encounters: a comparative study. In: Language Resources and Evaluation Conference (LREC 2012). Istanbul, Turkey (2012)
23. Nothdurft, F., Ultes, S., Minker, W.: Finding appropriate interaction strategies for proactive dialogue systems—an open quest. In: Proceedings of the 2nd European and the 5th Nordic Symposium on Multimodal Communication 2014, pp. 73–80. LiU Electronic Press, Tartu, Estonia (2015)
24. Okonogi, K., Wilcock, G., Yamamoto, S.: Nihongo WikiTalk no kaihatsu (Development of Japanese WikiTalk). In: Forum on Information Technology (FIT 2015). Matsuyama, Japan (2015). (in Japanese)
25. Paggio, P., Allwood, J., Ahlsén, E., Jokinen, K., Navarretta, C.: The NOMCO Multimodal Nordic Resource - Goals and Characteristics. In: Proceedings of Seventh International Conference on Language Resources and Evaluation (LREC 2010), pp. 2968–2973. Valletta, Malta (2010)
26. Reina, L.A., Robles, G., González-Barahona, J.M.: A preliminary analysis of localization in free software: How translations are performed. In: Open Source Software: Quality Verification, pp. 153–167. Springer (2013). DOI 10.1007/978-3-642-38928-3_11
27. Roturier, J.: Localizing Apps: A Practical Guide for Translators and Translation Students. Translation Practices Explained. Taylor & Francis (2015)
28. Toivio, E., Jokinen, K.: Multimodal Feedback Signaling in Finnish. In: Proceedings of the Fifth International Conference: Human Language Technologies - The Baltic Perspective, pp. 247–255. Tartu, Estonia (2012)
29. Unicode Consortium: Unicode Common Locale Data Repository. URL `http://cldr.unicode.org/`. Accessed: 2015-09-18
30. Wang, X., Zhang, L., Xie, T., Mei, H., Sun, J.: Locating need-to-externalize constant strings for software internationalization with generalized string-taint analysis. Software Engineering, IEEE Transactions on **39**(4), 516–536 (2013). DOI 10.1109/TSE.2012.40
31. Wilcock, G.: WikiTalk: A Spoken Wikipedia-based Open-Domain Knowledge Access System. In: Proceedings of the COLING 2012 Workshop on Question Answering for Complex Domains, pp. 57–69. Mumbai, India (2012)
32. Wilcock, G., Jokinen, K.: Multilingual WikiTalk: Wikipedia-based talking robots that switch languages. In: Proceedings of the 16th Annual SIGdial Meeting on Discourse and Dialogue. Prague (2015)
33. Wilcock, G., Laxström, N., Leinonen, J., Smit, P., Kurimo, M., Jokinen, K.: Towards SamiTalk: a Sami-speaking Robot linked to Sami Wikipedia. In: Proceedings of Seventh International Workshop on Spoken Dialogue Systems (IWSDS 2016). Saariselkä (2016)
34. World Wide Web Consortium: Localization vs. internationalization. URL `http://www.w3.org/International/questions/qa-i18n`. Accessed: 2015-09-18