In: *Proceedings of the 3rd Conference on Conversational User Interfaces (CUI)*, 2021, doi:10.1145/3469595.3469631 **POSTPRINT**
The official publication is available at dl.acm.org

Evaluating NLG-frameworks for multilingual surface realization in conversational assistants

Hanne Brockow
hanne.brockow@miele.com
Miele & Cie. KG
Gütersloh, Germany

Hendrik Buschmeier
hbuschme@uni-bielefeld.de
Digital Linguistics Lab, Bielefeld University
Bielefeld, Germany

ABSTRACT

Conversational voice assistants that are available in multiple countries need to be able to generate utterances in the language that their users speak. In open domains, in which messages can be variable, pre-writing and translating all utterances in advance is unfeasible because it is costly, error-prone, and inflexible when changes need to be made. Approaches to automatically generate multilingual surface forms of utterances have been developed in the field of Natural Language Generation (NLG), however, these are rarely used when developing skills for conversational voice assistants. In this paper, we describe an evaluation study that analyses the feasibility of integrating NLG surface-realization frameworks (SimpleNLG and RosaeNLG) into the development process of an existing commercial and multilingual (English, French, German, Italian, Spanish) home-automation skill, and compare it to a more traditional localization approach. The study uses methods and measures from human–computer interaction and software engineering, and takes into account the perspective of various stakeholders in the development process (conversation designers, language experts and developers).

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; • **Human-centered computing** → **Natural language interfaces**; *Empirical studies in HCI*; • **Software and its engineering** → *Domain specific languages.*

KEYWORDS

conversational agents, natural language generation, multilingual surface realization, internationalization, evaluation

ACM Reference Format:
Hanne Brockow and Hendrik Buschmeier. 2021. Evaluating NLG-frameworks for multilingual surface realization in conversational assistants. In *CUI 2021 – 3rd Conference on Conversational User Interfaces (CUI '21), July 27–29, 2021, Bilbao (online), Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3469595.3469631

CUI '21, July 27–29, 2021, Bilbao (online), Spain

ACM ISBN 978-1-4503-8998-3/21/07…$15.00
https://doi.org/10.1145/3469595.3469631

1 INTRODUCTION

Conversational voice assistants (e.g., for 'home automation' applications) that are available in multiple countries should be able to generate (and understand) utterances in the languages that their users speak. To that end, each 'message' that can be expressed as an utterance by the conversational assistant needs to be localized for each language. In contrast to standard internationalization-tasks (such as for user interfaces or user manuals of a single device) the number of utterances that a conversational assistant needs to be able to produce is not necessarily limited to a fixed set: in the home automation context, for example, new devices might be added to the system or new functionalities may become available (e.g., through the combination of multiple devices). Thus, simply translating all utterances of the conversational agent may not be possible.

A standard approach for generating an open set of utterances for conversational agents is to use templates in which certain parts are placeholders that are filled as needed. The utterances "The washing machine is ready." and "The oven is ready." could, for instance, be easily generated from an utterance template "The *{deviceType}* is ready." However, when generating sentences in multiple languages, using templates becomes more complex. Depending on the language, inserting a word in a placeholder such as *{deviceType}* may require further changes to other words for the utterance to remain grammatically correct. Consider the examples above in Spanish, where the words for washing machine and oven, "lavadora" and "horno", cannot simply be inserted for *{deviceType}*, as they have different grammatical genders, and therefore require the article and the adjective to be in agreement with them:

(1) ***La*** *lavadora* *está* *list**a**.*
the.DEF.F.SG washing machine.F.SG be.PRES.3SG ready.F.SG.
'The washing machine is ready.'

(2) ***El*** *horno* *está* *list**o**.*
the.DEF.M.SG oven.M.SG be.PRES.3SG ready.M.SG.
'The oven is ready.'

Similar morphological adjustments are needed when referring to more than one device:

(3) ***La**s* *lavadora**s*** *está**n*** *list**as**.*
the.DEF.F.PL washing machine.F.PL be.PRES.3PL ready.F.PL.
'The washing machines are ready.'

(4) ***Lo**s* *horno**s*** *está**n*** *list**os**.*
the.DEF.M.PL oven.M.PL be.PRES.3PL ready.M.PL.
'The ovens are ready.'

A simple technical solution that can be used for template-based language generation in multilingual conversational agents is the

```
{deviceTypeGender, select,
    MASCULINE{
        {deviceTypePlural, plural,
            one{El {deviceType} está listo}
            other{Los {deviceType} están listos}}}
    FEMININE{
        {deviceTypePlural, plural,
            one{La {deviceType} está lista}
            other{Las {deviceType} están listas}}}
    other{El {deviceType} está listo}}
```

Figure 1: ICU message example for the Spanish localization of the sentence "The *{deviceType}* is/are ready.", modeling grammatical agreement for gender and number.

message format[1] that is part of the ICU-framework [1], developed by the Unicode Consortium for internationalization of software and websites. Figure 1 shows an ICU message for the Spanish examples above.

Although generic internationalization frameworks such as ICU are suitable software tools for defining multilingual messages, their use is costly: each new utterance that is added to a multilingual conversational assistant must be translated into each language, converted into an adequate ICU message, and tested for grammatical correctness – a task that requires expert knowledge of the languages as well as of the ICU format, and the conventions used for defining the grammatical relations. Depending on the complexity of an utterance and the grammatical complexity of a language, message templates can quickly become unwieldy and thus difficult to maintain.

More advanced approaches to the problem of realizing grammatically correct utterances (a task called 'linguistic surface realization') are developed in the field of Natural Language Generation (NLG [2]). The idea is that the linguistic knowledge of a language is represented once – as part of the surface realization framework – and needs not be encoded (and repeated) in each message definition. Messages can thus be specified in an abstract form, the realization algorithm will automatically take care of the linguistic details when generating utterances. In practice, such frameworks often support a combination of template and rule-based generation [7].

In this paper, we evaluate whether the use of automatic surface realization frameworks is a viable alternative to the standard template-based localization approaches used in multilingual conversational user interface design. We do not approach this question from an end-user's perspective, but rather from a development perspective, and take technical as well as broader organizational aspects into account. Apart from a general evaluation of the applicability of surface realization frameworks for the task (i.e., can it generate a representative sample of the sentences needed?), we use methods from the field of human–computer-interaction (HCI), and measures from software engineering to evaluate the 'usability'[2] of such frameworks. This is done from the perspective of various stakeholders involved in the design and implementation of the commercial, multilingual Miele voice assistant skill.

[1]ICU's *MessageFormat*: https://unicode-org.github.io/icu/userguide/format_parse/messages/ (accessed 2021-04-05).

[2]Usability is put in quotes here to highlight that it does not mean usability for end-users but for developers.

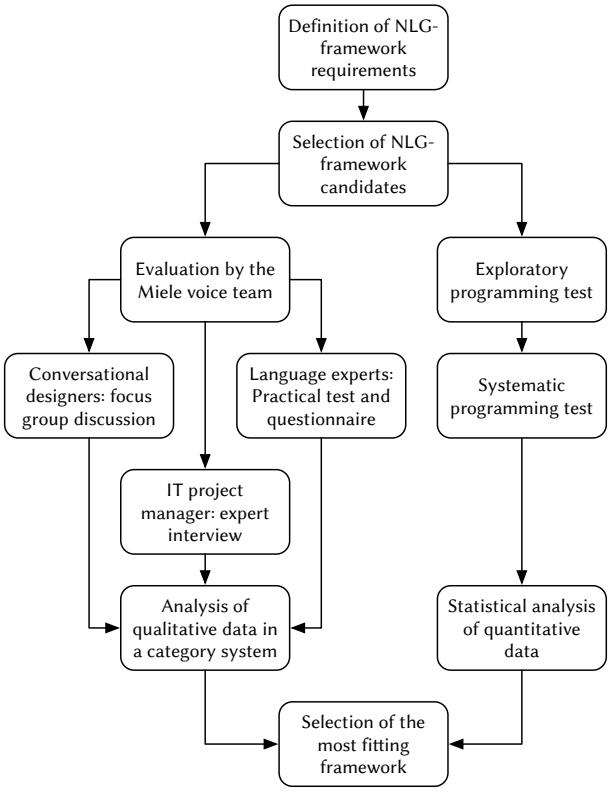

Figure 2: Procedure for evaluating the NLG-frameworks for the Miele voice assistant skill. To select a suitable framework, technical as well as broader organizational perspectives were taken into account.

2 METHODS

To find suitable natural language generation frameworks for the voice assistant skill, we formulated requirements and screened frameworks through an online search. We then selected the two most promising frameworks for in-depth analysis. We performed an exploratory and a systematic programming test and carried out usability tests involving the subgroups of the Miele voice team. Figure 2 provides an overview of the procedure.

Requirement analysis and framework selection. We began the selection process by establishing the following requirements for a surface realization framework suitable for the voice skill: (i) it should be able to generate text in all five languages that are currently supported (English, French, German, Italian, Spanish); (ii) it should be extensible so that more languages and additional features can be integrated; (iii) it should use a well-established programming language; and (iv) it should be published under a license that permits commercial use. We carried out an online search for candidates and identified 18 potential NLG and/or surface-realization frameworks (a full listing is provided in the Supplementary Material) that we then filtered using our requirements. The two most promising

frameworks, SimpleNLG [3] and RosaeNLG [6], were selected for a detailed evaluation and comparison with the currently used ICU message format.

Programming tests. As a first analysis, the first author conducted an *exploratory* programming test to explore whether the two frameworks could handle the linguistic aspects and features that are necessary for the voice skill. Several utterances from the skill were modeled and generated with both frameworks in all five languages. Utterances which needed verb and adjective agreement were particular challenges. We did not include the ICU format in this part of the assessment, since it is already used in the voice skill and is thus capable of handling the requirements.

Following this, the first author conducted a *systematic* programming test in which eleven example sentences from the voice skill were modeled and generated from scratch in all five languages, using the selected frameworks as well as the ICU format. In this test, we collected quantitative measures such as the time needed to implement each sentence with each approach, how many lines of code were written, and how many errors and failed attempts occurred before a correct sentence could be produced (see table 1). The data collected was then analyzed statistically.

Usability tests. The second part of the study focused on whether – and how well – the NLG-frameworks could be integrated into the development process of the voice team. For this, we used methods from HCI, namely user experience and usability testing, and collected mostly qualitative data using different collection techniques, such as interviews, questionnaires, and focus group discussions. Subgroups of the voice team (conversational designers, developers, and language experts; see Figure 2) evaluated if the frameworks can fulfill the needs of the different groups according to their field of expertise. Since these stakeholders would all use the framework for different purposes, we chose a different evaluation method for each subgroup to evaluate the functionalities that are relevant for them. For example, the language experts took part in the most thorough testing, because such a framework influences their future work the most.

Three *conversational designers* from the team listened to a 25-minute presentation about the frameworks and participated in a focus group discussion, in which they analyzed the conceptual advantages and disadvantages of each, and prioritized the aspects that they would ideally want. Following this, an IT project manager, who represented the *developers* working on the skill, listened to a very similar short presentation about the technical aspects of the frameworks and took part in a semi-structured interview. During the interview, it was discussed for each framework which possible technical problems it might have and how it could be integrated into the skill. Finally, three *language experts* from the team first performed a practical test, in which they implemented three linguistically challenging sentences from the voice skill with all three frameworks in all five languages. Afterwards, they evaluated the frameworks and their experience using a structured questionnaire consisting of 32 open and 57 closed questions about different aspects of framework performance (the questionnaire is provided in the Supplementary Material). Questions centered on how intuitively each framework can be used, the number of errors, the time required, and how much programming, language and linguistic knowledge is needed to interact with each framework. The transcript of the focus group discussion, as well as the transcript of the interview and the questionnaire were then analyzed qualitatively with a category system in which relevant aspects are generalized and then sorted into categories (e.g., advantage/disadvantage of ICU/RosaeNLG/SimpleNLG/NLG-frameworks) [4].

3 RESULTS

The different tests carried out demonstrated that using automated NLG-frameworks for applications such as the voice assistant skill for home automation do indeed have several benefits in comparison to standard localization approaches such as the ICU-framework. Most importantly, users of NLG-frameworks need less knowledge about a specific language, since these frameworks automatically take care of grammatical agreement and morphological forms of words. Further advantages include the ability to automatically generate lists with dynamic elements and a much-simplified error detection and correction system, as utterance implementation is supported by integrated development environments. Finally, the specification of utterances is more concise than in the ICU-framework, since ICU-messages are often long and difficult to read due to deeply nested structures.

The ICU-framework, however, also offers certain advantages. First of all, ICU enables greater flexibility as utterances can be implemented freely without restrictions. Furthermore, ICU is a commonly used localization format (e.g., for websites or software), and many translation and localization agencies are already familiar with it. Additionally, no programming skills and less linguistic knowledge (but more language-expertise) is needed to work with the format. Finally, it should be mentioned that the ICU format and libraries are backed by a large international organization (the Unicode Consortium), whereas the evaluated NLG-frameworks – though being open source as well – are mainly developed by a small number of persons. This makes their long-term availability and support less certain.

Analyzing the two NLG-frameworks, we found that RosaeNLG performed better than SimpleNLG in most of the tests we carried out. This was mainly due to the template-based structure and flexibility that RosaeNLG offers, which enables generation of clauses both in an automatic manner and as 'canned text' [5]. Furthermore, RosaeNLG allows for a higher degree of variation and, probably due to the simpler syntax, a faster implementation of new utterances, resulting in more concise utterance specifications, as well as less implementation errors in the systematic programming test when compared to the implementation of utterances with SimpleNLG (see Table 1; these differences are statistically significant). Nevertheless, SimpleNLG is not without advantages. The framework offers more features, such as the ability to automatically generate negations of sentences and clauses. In addition, the structure in which the user defines constituents and phrases of utterances is very well designed. In general, both programming tests revealed a small number of errors in both frameworks. The higher degree of control that RosaeNLG offers users made it easier to prevent the generation of grammatically incorrect utterances than was the case with SimpleNLG.

Table 1: Quantitative results from the systematic programming tests implementing eleven utterances in all five languages. The table shows the (min/max/mean) time needed to implement an utterance in a framework, the number of lines of code of the utterance definition, the number of failed attempts until a correct sentence could be generated, and the number of errors that needed to be corrected from the first to the last attempt of each sentence.

Framework	Time (min)		Lines of Code		Failed Attempts		Errors	
	min/max	*mean (sd)*	*min/max*	*mean (sd)*	*min/max*	*mean (sd)*	*min/max*	*mean (sd)*
RosaeNLG	1/30	2.71 (4.07)	4/21	8.8 (3.82)	0/7	0.49 (1.2)	0/2	0.31 (0.51)
SimpleNLG	2/30	6.53 (5.9)	6/26	11.8 (4.48)	0/10	1.6 (2.17)	0/3	0.87 (0.82)
ICU	1/10	2.87 (2.15)	1/30	13.58 (7.78)	0/2	0.15 (0.49)	0/3	0.29 (0.63)

4 CONCLUSIONS

When examining the different test methods, we observed that they produced very similar results. This indicates that they are an effective way to evaluate NLG-frameworks for use in real-world applications. Through an online search, a pre-selection of a wide variety of NLG-frameworks became available. The definition of requirements enabled the narrowing down of the candidates to two promising frameworks. The exploratory programming test revealed that RosaeNLG as well as SimpleNLG are, generally, both viable alternatives to the established ICU-based approach, and are capable of generating voice assistant utterances in all five languages. Even though the conversational designers, the IT project manager, and the language experts of the Miele voice team all focused on different aspects of the frameworks (and all had different prerequisites), they all ranked them similarly, and found comparable advantages and disadvantages. This shows that a study with diverse methods and a small number of participants can lead to reliable results. The systematic as well as the exploratory programming test confirmed the assessment of the team members, and revealed statistically significant differences between RosaeNLG and SimpleNLG, as well as comparable results between RosaeNLG and ICU.

Due to the variety of evaluation methods used and an evaluation approach that takes the perspectives of different stakeholders into account, the decision of whether a certain NLG-framework could be used in the development process of a specific voice assistant appears solidly founded. From a natural language processing and user experience point of view (with users' of the framework being developers/designers/language experts), it was possible to identify the most suitable framework for the task. Furthermore, involving multiple stakeholders in the framework evaluation process has the benefit that team members are already familiar, to some degree, with the framework, and will likely support its introduction in the development process.

It should be noted, however, that a limitation of the current study is that economic and business aspects of the choice of framework were not analyzed – although these clearly play a significant role in decision making as well.

5 SUPPLEMENTARY MATERIAL

A listing of the 18 NLG-frameworks considered in the initial screening and an English translation of the questionnaire used in the practical test with the group of language experts are available as supplementary material: http://doi.org/10.6084/m9.figshare.14680467

REFERENCES

[1] Unicode Consortium. 2021. *ICU – International Components for Unicode.* Unicode Consortium, Mountain View, CA, USA. http://site.icu-project.org/

[2] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170. https://doi.org/10.1613/jair.5477

[3] Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009).* Association for Computational Linguistics, Athens, Greece, 90–93.

[4] Philipp Mayring. 2015. *Qualitative Inhaltsanalyse.* Beltz, Weinheim, Germany.

[5] Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems.* Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511519857

[6] Ludan Stoecklé. 2021. *RosaeNLG.* LF AI & Data Foundation, San Francisco, CA, USA. https://rosaenlg.org/

[7] Kees van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics* 31 (2005), 15–23. https://doi.org/10.1162/0891201053630291