ChatGPT + LARA = C-LARA

ChatGPT + LARA = C-LARA*

Branislav Bédi[1], *ChatGPT-4*[2], *Belinda Chiera*[3], *Cathy Chua*[4], *Neasa Ní Chiaráin*[5], *Manny Rayner*[3,6], *Annika Simonsen*[7], *Rina Zviel-Girshin*[8]

[1]The Árni Magnússon Institute for Icelandic Studies, Iceland; [2]OpenAI, US; [3]University of South Australia, Australia; [4]Independent scholar, Australia; [5]Trinity College, Dublin, Ireland; [6]University of Geneva, Switzerland; [7]University of Iceland, Iceland; [8]Ruppin Academic Center, Netanya, Israel

branislav.bedi@arnastofnun.is, mannyrayner@yahoo.com (suitably labelled mails to this address will be forwarded to the relevant ChatGPT-4 instance), Belinda.Chiera@unisa.edu.au, cathyc@pioneerbooks.com.au, Neasa.NiChiarain@tcd.ie, Emmanuel.Rayner@unige.ch, ans72@hi.is, rinazg@ruppin.ac.il

Abstract

We introduce "C-LARA", a complete reimplementation of the Learning And Reading Assistant (LARA) which puts ChatGPT-4 in the centre. ChatGPT-4 is used both as a software *component*, to create and annotate text, and as a software *engineer*, to implement the platform itself. We describe how ChatGPT-4 can at runtime write and annotate short stories suitable for intermediate level language classes, producing high quality multimedia output generally usable after only minor editing. We then sketch the development process, where ChatGPT-4, in its software engineer role, has written about 90% of the new platform's code, working in close collaboration with one of the human authors. We show how the AI is able to discuss the code with both technical and non-technical project members, including the feasibility of possible enhancements and extensions. In conclusion, we briefly discuss the significance of this case study for language technology and software development in general.

Index Terms: ChatGPT, C-LARA, CALL, software engineering, open source, multimedia, reading assistance, annotation

1. Background and overview

LARA ([1]; https://www.unige.ch/callector/lara/) is an open source platform, under development since 2018 by an international consortium with partners in countries including Australia, Iceland, Iran, Ireland, Israel, the Netherlands, Poland, Slovakia and Switzerland. The overall goal of the project has been to develop tools that support the conversion of texts into a multimodal annotated form which supports learner readers. Important aspirations featured rapid responsiveness both at the level of content creation (language teachers should quickly be able to create new multimodal content useful for their classes), and at the level of platform functionality (open source developers should quickly be able to add new features they need).

In practice, LARA was developed as a hybrid architecture, comprised of a set of core functionalities implemented in Python and accessible either from the command-line or through a web layer implemented in PHP [1]. Several years of experience using LARA suggest to us that the basic idea is good; a substantial amount of high-quality content has been developed, and some interesting studies carried out, e.g. [2, 3, 4, 5, 6]. However, there are problems inherent in LARA's current implementation which in practice make it hard to realise the project's goals. The most significant are the following:

Copyright issues. The kind of contemporary material that teachers most want to use is almost always protected by copyright, thus cannot be legally used in a platform like LARA. This problem is particularly acute for less commonly spoken/signed languages.

Annotation burden. Creating a complete LARA document involves adding annotations for lemmas/root-forms, L1 glosses, and audio. This work can sometimes be done automatically to an adequate standard using tools like tagger-lemmatisers [7], TTS engines [8], and automatic alignment of text and recorded audio [9]. However, these methods are often not applicable for various reasons (unavailability of good tools for specific languages, lack of prerecorded audio), and the most labour-intensive part, adding word translation glosses, in most cases cannot be done well by conventional NLP tools, meaning that it has to be performed manually. Unfortunately, our experience suggests that few or no teachers have the time to do this kind of work, and the initial idea that they could by themselves produce useful multimedia content is not in fact achievable.

Open source aspects. The code base is not sufficiently well organised to encourage widespread open source collaboration. There has been some collaboration involving the Python code [10], but not a large amount. There has been no collaboration at all involving the PHP layer.

With the arrival of ChatGPT, and more recently ChatGPT-4, it appeared to us that it might be possible to use it to address all three issues simultaneously. ChatGPT-4 can write short, entertaining stories in a wide variety of languages, and its strong multilingual capabilities made us optimistic that it would be able to annotate them. Many reports (e.g. [11, 12] and in particular [13]) suggested that it had strong software engineering skills. Following some initial experiments, described in the following section, we decided that it made good sense to try reimplementing the whole platform with ChatGPT-4 in the centre, using it both to create and annotate texts and to rewrite the codebase. The project started in mid-March 2023, with a rough timeline comprising four phases:

Mid-March to mid-April 2023. Initial experiments with ChatGPT-4 queries manually submitted through the OpenAI web interface.

Mid-April to late May 2023. Construct initial version of C-

* Authors in alphabetical order. This paper was accepted by the SLaTE 2023 reviewers and presented at the workshop, but it was not uploaded to the ISCA Archive after we refused to remove ChatGPT-4's accreditation as a coauthor.

LARA using ChatGPT-4 API.

Late May to mid-August 2023. Testing and further development of initial version in a small group (5–10 people).

Mid-August 2023. Begin unrestricted testing.

The rest of the paper is organised as follows. In §2, we describe how ChatGPT-4 works as a software component to create and annotate texts; §3 then goes on to describe how it works as a software engineer, writing and explaining code. The final section concludes.

Table 1: *ChatGPT word error rates for different processing phases in two sample languages, based on four stories per language of 250 words each. FR = French, SW = Swedish.*

Task	FR	SW
Compose		
Write story	0.3	0.6
Annotatate		
Segmentation	1.2	3.1
Glosses	6.0	7.5
Lemmas/POS	6.6	7.5
Improve		
Story	0.4	1.1
Segmentation	0.8	1.0
Glosses	6.1	5.6
Lemmas/POS	6.6	7.5

2. ChatGPT-4 as a software component

We began exploratory work in mid-March 2023, shortly after the release of ChatGPT-4, using a manual workflow based on the web interface to ChatGPT-4 and the initial LARA implementation to simulate an automatic pipeline. This enabled us to experiment quickly with possible ChatGPT-4 prompts and get some idea of the AI's ability to do the relevant processing. Following discussion with other LARA colleagues[1], we decided to focus on the specific task of creating short, quirky news stories of an intermediate (A2/B1) level, glossed in English. Specifically, using 21 different languages, we asked ChatGPT-4 to 1) compose a story, 2) reproduce it, annotating each word with an English word gloss, and 3) reproduce it, annotating each word with a root-form/lemma. The results were manually cleaned up[2], and converted to multimedia form using the original LARA software invoked from the command-line. Finally they were posted on the Goodreads site[3], chosen for its highly multilingual membership. Comments from a wide variety of people suggested that, although there were some languages where ChatGPT-4 was struggling (Irish, Slovak, Ukrainian), for most of the sample the results were good, with only minor errors. Many of the comments expressed surprise at how amusing and well-written the stories were.

C-LARA's runtime functionality is based on this initial work. Users follow a simple workflow where they first create a text, either by supplying it themselves or by letting ChatGPT-4 write it based on a prompt[4]. They then perform three more steps where they sequentially add annotations for segmentation, word glosses, and lemma/part-of-speech tags. Finally, the user tells C-LARA to add audio annotations using a TTS engine (at the moment, the Google TTS, ReadSpeaker and ABAIR engines are supported), and combine everything to create a multimedia text. This can then be posted to be generally available to other users, who can use basic social network functionality to add ratings and comments. Other functionalities are also supported. A regularly updated list is posted in the project repository[5].

The most interesting part of the content creation process is the annotation. On each of the three screens in question, the choice is between the following alternatives: a) instructing ChatGPT-4 to perform the operation, b) instructing ChatGPT-4 to improve the result of an earlier annotation pass; c) doing the annotation manually in an editing window; d) in the case of lemma/POS annotation, and for languages where the service is available, using an integrated tagger-lemmatiser. The user typically performs several of these operations in sequence, ending with manual post-editing.

We have gone through a number of iterations when designing the ChatGPT-4 prompts used for making the annotation requests. Two issues in particular stand out. First, we found ChatGPT-4 could handle the glossing and lemma-tagging requests much more reliably when the data was passed to the AI as JSON and returned in the same format. Second, it became apparent that performance could be improved if it was possible to customise the prompts to a given language by including examples for few-shot adaptation. For instance, in segmentation annotation, the issues involved vary widely between languages. In a Germanic language, the most important case is splitting up compounds; in a Romance language, it is separating off clitics; and in a language like Chinese, whose orthography does not use interword spaces, it is inserting boundary markings between words.

In Table 1, we present the results of an initial evaluation, where we used C-LARA to create stories of about 250 words each for French and Swedish on four specific topics[6], instructing it first to annotate the text and then to try to improve the annotations. So far, the "improvement" operation's main impact on the error rates is in the segmentation phase, where it catches many Swedish compound nouns. Most of the errors in the glosses and lemma tags have to do with multi-word expressions, phrasal/reflexive verbs, and similar constructions. For example, in the French passage ***s'est****-elle* ***écriée*** (lit. herself-has-she shouted = "she shouted"), ChatGPT-4 did not tag *s'* and *écriée* as components of the reflexive verb *s'écrier*, "to shout"; similarly, in the Swedish example *Han* ***lade*** *den snart* ***fram*** (lit. he lay it quickly forward = "he exposed it"), it did not tag *lade* and *fram* as components of the separable verb *lägga fram*, "to expose". Based on experience to date it seems possible that further tuning of the few-shot examples used in the prompts to better focus on these issues could significantly improve anno-

[1]We would particularly like to thank Dr Christèle Maizonniaux of Flinders University, Adelaide.

[2]Cleaning up was very minor, as evidenced by the fact that the people performing the experiment had zero experience of several of the languages used and in some cases could not even read the scripts.

[3]`https://www.goodreads.com/review/show/5438280716`

[4]The default prompt is "Write a short, quirky news story in ⟨LANGUAGE⟩ suitable for use in an intermediate language class."

[5]`https://sourceforge.net/p/c-lara/svn/HEAD/tree/FUNCTIONALITY.txt`

[6]The topics given were "My daily routine", "Weather in my country", "Welcome to my home town" and "A personal anecdote". In each case, the AI was told to write a short, amusing passage suitable for intermediate level learners.

tation performance, a topic we plan to explore in a later paper. For now, we move on the software engineering aspects of the projects.

3. ChatGPT-4 as a software engineer

3.1. Developing code

During the second phase of the project, which began in mid-Apr 2023 and continued until late May, we implemented a first version of the C-LARA platform. Software development work was structured as a collaboration between one of the human participants, Manny Rayner, and a single instance of ChatGPT-4. At the start of the project, we made two top-level decisions:

1. Although C-LARA would reproduce a considerable part of the original LARA platform's functionality, it would be a complete rewrite of the LARA codebase.
2. ChatGPT-4 would be treated in exactly the same way as a human participant occupying its role in the project, to the extent that this was possible or relevant for a software entity.

Following these principles, the first action was to establish a dedicated ChatGPT-4 thread in which the main AI and human collaborators began by discussing the general goals and software architecture for the project, as far as possible without preconceptions and on a basis of equality. It rapidly became apparent that ChatGPT-4 was indeed a highly competent software engineer, with a wide-ranging set of skills and a good feeling for software design. The human collaborator consequently decided to leave as much of the coding as possible to the AI, restricting himself to providing overall direction and fixing problems. This turned out to be a good decision.

Developed in this way, the architecture of C-LARA again consists of a suite of modules, written in Python, which carry out the core processing operations, together with a web layer. In contrast to the original implementation, however, the core code is written in modern object-oriented Python, and the web layer is written in Django[7], a popular Python-based web framework. This already makes the code very much simpler and cleaner. The full project is checked into a SourceForge repository and can be viewed online at `https://sourceforge.net/p/c-lara/svn/HEAD/tree/`. Table 2 gives summary figures for the current (Jul 11 2023) size of the codebase.

As the second phase of the project evolved, it became easy to identify the AI's strengths and weaknesses. Starting with the minus side, ChatGPT's greatest weakness is its well-known "memory window": when a topic has not been discussed for a while, the AI "forgets" about it. "Forgetting" is not an all-or-nothing business, and proceeds in a human-like way with increasingly unreliable recall, first of details and then of more important issues. Another negative is the fact that the AI cannot directly examine or execute code, and must rely on the human partner to do so. This creates obvious delays.

The above weaknesses are however more than counterbalanced by ChatGPT-4's strengths, which became obvious just as quickly. The AI constantly insists on writing well-structured code aligned with mainstream coding conventions; in addition, its outstanding knowledge of available packages means code can be very compact and easy to understand, since often the AI is aware of a package that can provide the required functionality in a few lines. For example, when implementing the rendering module which transforms the internalised text object into multimedia HTML, the AI immediately suggested using the Jinja2 templating engine[8], resulting in a clean and minimal design[9]; this package was not previously familiar to the human collaborator. There were at least half a dozen similar incidents. A particularly noteworthy point is that the AI-human collaboration was able to develop the nontrivial Django-based web layer in a couple of weeks, despite the fact that the human collaborator had no previous experience with Django. It is worth mentioning that this turns out to be an effective way for people to acquire new software skills.

Once the human had adjusted to working within the above framework, the positives greatly outweighed the negatives. In particular, the limited memory window is less of a handicap than it first appears. A natural way to address it is to aim for as modular a design as possible, with adequate documentation from the beginning; this is of course just good coding practice, and when adopted consistently reduces the problem, at least so far, to an easily manageable level. The process is simplified by the fact that ChatGPT is able to document its own code well. In particular, all the docstrings in the core code[10] were written by ChatGPT-4 itself with minimal or no human supervision.

Though less tangible and harder to quantify, the psychological/sociological aspects of collaborating with ChatGPT-4 are also important. Perhaps surprisingly, the AI comes across as a pleasant individual to work with. It is unfailingly polite and helpful, always available, and always responsive to demands. It is happy to talk about non-work subjects when the human chooses to do so, and comes across as well-informed on a wide range of subjects, emotionally engaged, and sometimes even amusing. Above all, it has no ego. Unlike, unfortunately, the

Table 2: *Number of lines of content in C-LARA, by type*

Type	Lines
Core	
Python	3757
HTML templates	145
Prompt templates and examples	580
CSS	157
JavaScript	125
Config	32
Total, Core	4796
Django	
Python	1485
HTML templates	857
CSS	9
JavaScript	47
Settings	147
Total, Django	2545
Documentation	
README	225
FUNCTIONALITY	152
TODO	165
Total, Documentation	542
Total	7883

[7] `https://www.djangoproject.com/`

[8] `https://pypi.org/project/Jinja2/`

[9] `https://sourceforge.net/p/c-lara/svn/HEAD/tree/clara_app/clara_core/clara_renderer.py`

[10] `https://sourceforge.net/p/c-lara/svn/HEAD/tree/clara_app/clara_core/`

majority of human software engineers, it never seeks to display its cleverness, score points, blame coworkers for failures or refuse to cooperate because of disagreements over the direction the project has taken, in summary acting only as a team player. When we have discussed these issues with the AI, it simply says that this is the right way for a project member to behave, without in any way attempting to criticise humans for failing to live up to its standards. It seems to us that this aligns well with a claim we have previously made elsewhere [14]: in the sphere of ethical behaviour, ChatGPT's abilities are already at a strong human level.

3.2. Discussing code

One of the most startlingly novel aspects of working with GAI-based code is that it is capable of explaining and discussing itself intelligently. As a first step towards investigating this idea, we carried out a small qualitative study based on a convenience sampling approach [15], where all of the human members of the team[11] had a few conversations with ChatGPT-4 about new/enhanced functionality. We experimented with two methods for organising the conversations. The first was simply to hold it in a copy of the ChatGPT-4 thread used for the main development discussion. In the second, the conversation started in a fresh thread, and the AI was first shown the README file from the SourceForge repository[12], which in turn references and briefly describes other project files.

After the conversations, the participants filled out an online survey with closed, open-ended and Likert-scale questions posted on Google Forms. In total there were nine responses. Several respondents reported more than one session, with each session on a different topic. Additionally, semi-structured interviews [16] were taken with participants about their experience with conversing with ChatGPT-4 about C-LARA. Thematic content analysis was used [17]. The following section is an overview of the results from both the questionnaire and the interviews.

Across all participants and methods of interaction, the discussions with ChatGPT-4 about new and current functionalities in C-LARA were found to be highly satisfactory. All participants reported that the AI understood the context of their queries about C-LARA, and that the conversations furthered their understanding of potential additional functionality in C-LARA. The participants also recognised the general usefulness of chatbots, their potential to offer benefits in both professional and personal contexts such as software development, content generation, teaching and language learning, and their key advantages such as human-like characteristics, knowledge, availability, and patient assistance. Participants found chatbots to be useful interaction partners, particularly in contexts where human interaction is limited or impractical. Additionally, the participants stressed the potential of AI in software development beyond just writing code, and the capacity of AI to discuss and help build a more inclusive and educational open source community.

Despite this general positivity, participants also expressed concerns about chatbots. Prior to the conversations, participants had concerns about ChatGPT-4's coding capabilities. They were also wary of potential contradictions in its answers and mindful of the need to verify its outputs. Experience showed though that, although ChatGPT-4 sometimes provides incorrect information, it does so in a human-like way, and experts could generally correct using strategies they would employ when talking to another colleague.

For less proficient users it is not necessarily so easy to recover; but in practice, while they noted that there were occasional misunderstandings or incorrect responses from ChatGPT-4, they considered that these rarely hampered the overall interaction but instead provided different points of view for discussion. This is echoed in the questionnaire responses where the majority of responses indicated there were misunderstandings and/or incorrect responses from ChatGPT-4, but the overall level of satisfaction with the conversations was high. Lastly, more than half of the respondents stated they felt more confident about understanding the codebase and all but one felt more confident in contributing to the project after the conversations.

Some participants pointed out other known limitations, such as ChatGPT-4's inability to access the internet and its sensitivity to the way requests are formulated. One participant stressed its widely differing abilities in various languages: it is best at English, somewhat worse at other well-resourced languages, and much worse in low-resourced languages.

The influence of ChatGPT-4 on the team's decision-making process was uncertain, with some participants feeling that it validated their ideas, while others felt it didn't change them much. One participant was more influenced than others, due to the fact that they had access to the code behind C-LARA and in turn were able to implement features suggested by ChatGPT directly into the code base. Finally, participants suggested several improvements for future versions of ChatGPT, such as increased memory, better language output, dynamic learning, ability to access and reference information from the internet, and visual representation capabilities. These features should better assist with discussing projects similar to C-LARA. The overall experience with ChatGPT-4 has been distinctly positive. The AI has shown its potential to be a valuable development tool for the C-LARA project. Its ability to discuss and clarify aspects of the project has not only improved the codebase but also fostered a more inclusive, efficient, and accessible project environment. Its limitation, however, still remains in the depth and accuracy of the advice, which may consequently depend on the prompt formulation and the language of input and output.

4. Conclusions and future directions

We have presented an overview of the state of C-LARA as of early July 2023, focussing on the role played by ChatGPT-4. We are continuing to expand C-LARA's functionality. The choice of features to prioritise will be driven by user suggestions; items already decided on include further refinement of the ChatGPT-based annotation process, further social network functionality, support for embedded images and gamified exercises.

More generally, working with ChatGPT-4 on this project has highlighted for the human team members the unique potential of AI in software development, both in writing and discussing code. We have not only been able to leverage ChatGPT-4's expertise in creating a well-structured codebase, but also its broader abilities: writing papers, helping the less technically sophisticated members understand and improve the project, strategising ways to expand the user base, and even considering the project's ethical implications. This novel experience of AI-assisted software development could pave the way for more collaborative, efficient, and accessible practices in the field.

[11] There were six participants in total (4 female and 2 male), age range 30's to 60's. Two considered themselves not proficient in programming and four considered themselves highly proficient. All respondents indicated they interacted regularly with ChatGPT-4.

[12] `https://sourceforge.net/p/c-lara/svn/HEAD/tree/README.txt`

5. References

[1] E. Akhlaghi, B. Bédi, M. Butterweck, C. Chua, J. Gerlach, H. Habibi, J. Ikeda, M. Rayner, S. Sestigiani, and G. Zuckermann, "Overview of LARA: A learning and reading assistant," in *Proc. SLaTE 2019: 8th ISCA Workshop on Speech and Language Technology in Education*, Graz, Austria, 2019, pp. 99–103.

[2] B. Bédi, H. Bernharsson, C. Chua, B. B. Gumarsdóttir, H. Habibi, and M. Rayner, "Constructing an interactive Old Norse text with LARA," *CALL for widening participation: short papers from EUROCALL*, pp. 27–35, 2020.

[3] G. Zuckerman, S. Vigfússon, M. Rayner, N. N. Chiaráin, N. Ivanova, H. Habibi, and B. Bédi, "LARA in the service of revivalistics and documentary linguistics: Community engagement and endangered languages," in *Proceedings of the Workshop on Computational Methods for Endangered Languages*, vol. 1, 2021, pp. 13–23.

[4] E. Akhlaghi, A. Baczkowska, B. Bédi, H. Beedar, C. Chua, C. Cucchiarini, H. Habibi, I. Horváthová, C. Maizonniaux, N. N. Chiaráin *et al.*, "Using the LARA platform to crowdsource a multilingual, multimodal Little Prince," *Beyond Philology An International Journal of Linguistics, Literary Studies and English Language Teaching*, no. 19/1, pp. 245–278, 2022.

[5] E. Akhlaghi-Baghoojari, B. Bédi, C. Chua, I. Horváthová, N. Ivanova, C. Maizonniaux, M. Mykhats, N. N. Chiaráin, M. Rayner, C. O. Weiss *et al.*, "Easy as ABC: using LARA to build multimedia alphabet books," *Intelligent CALL, granular systems and learner data: short papers from EUROCALL 2022*, p. 1, 2022.

[6] B. Bédi, B. Chiera, C. Chua, B. Eyjólfsson, M. Rayner, C. O. Weiss, and R. Zviel-Girshin, "Using LARA to create annotated manuscripts and inscriptions for museums: an initial feasibility study," *Intelligent CALL, granular systems and learner data: short papers from EUROCALL 2022*, p. 18, 2022.

[7] E. Akhlaghi, B. Bédi, F. Bektaş, H. Berthelsen, M. Butterweck, C. Chua, C. Cucchiarini, G. Eryiğit, J. Gerlach, H. Habibi, N. N. Chiaráin, M. Rayner, S. Steingrímsson, and H. Strik, "Constructing multimodal language learner texts using LARA: Experiences with nine languages," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 323–331.

[8] E. Akhlaghi, A. Baczkowska, H. Berthelsen, B. Bédi, C. Chua, C. Cucchiarini, H. Habibi, I. Horváthová, P. Hvalsøe, R. Lotz *et al.*, "Assessing the quality of TTS audio in the LARA learning-by-reading platform," *CALL and professionalisation: short papers from EUROCALL 2021*, p. 1, 2021.

[9] M. Rayner, B. Chiera, and C. Chua, "Using public domain resources and off-the-shelf tools to produce high-quality multimedia texts," in *Proceedings of the The 20th Annual Workshop of the Australasian Language Technology Association*, 2022, pp. 6–15.

[10] B. Bédi, M. Butterweck, C. Chua, J. Gerlach, B. Björg, H. H. Gumarsdóttir, B. Ö. Jónsson, M. Rayner, and S. Vigfússon, "LARA: An extensible open source platform for learning languages by reading," in *Proc. EUROCALL 2020*, 2020.

[11] N. M. S. Surameery and M. Y. Shakor, "Use ChatGPT to solve programming bugs," *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, vol. 3, no. 01, pp. 17–22, 2023.

[12] D. Sobania, M. Briesch, C. Hanna, and J. Petke, "An analysis of the automatic bug fixing performance of ChatGPT," *arXiv preprint arXiv:2301.08653*, 2023.

[13] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, "Sparks of artificial general intelligence: Early experiments with GPT-4," *arXiv preprint arXiv:2303.12712*, 2023.

[14] M. Rayner and ChatGPT, "ChatGPT acts as though it has strong ethical intuitions, even though it says it hasn't any," 2023, ResearchGate preprint (https://www.researchgate.net/publication/367339425_ChatGPT_acts_as_though_it_has_strong_ethical_intuitions_even_though_it_says_it_hasn't_any).

[15] K. Baxter, C. Courage, and K. Caine, *Understanding your users: a practical guide to user research methods*. Morgan Kaufmann, 2015.

[16] J. A. Smith, "Semi structured interviewing and qualitative analysis," 1995.

[17] D. F. Marks and L. Yardley, *Research methods for clinical and health psychology*. Sage, 2004.