



## King's Research Portal

### *Document Version*

Publisher's PDF, also known as Version of record

[Link to publication record in King's Research Portal](#)

### *Citation for published version (APA):*

Msosa, Y. J. (2019). FCIG Grammar Evaluation: A Usability Assessment of Clinical Guideline Modelling Constructs. In *IEEE Symposium on Computers and Communications IEEE Xplore*.

### **Citing this paper**

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

### **General rights**

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

### **Take down policy**

If you believe that this document breaches copyright please contact [librarypure@kcl.ac.uk](mailto:librarypure@kcl.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# *FCIG Grammar Evaluation: A Usability Assessment of Clinical Guideline Modelling Constructs*

Yamiko Joseph Msosa

Dept. of Computer Science, University of Cape Town / Dept. of Biostatistics & Health Informatics, King's College London

Email: ymsosa@cs.uct.ac.za;yamiko.msosa@kcl.ac.uk

**Abstract**—Computer-interpretable guidelines (CIGs) support application of clinical practice guidelines (CPGs) to improve clinical care delivery. But CPGs frequently change as the science and technology behind clinical practice improves, thereby necessitating changes in associated CIGs. This work aims to address CIG evolution challenges that arise due to changing CPGs. Specifically, this work evaluates the usability of *FCIG*, the grammar of a novel CIG modelling language, that enables computational management of CPG changes. The results showed that novice and experienced CIG modellers perceived *FCIG* as useful, necessary and usable for modelling evolving CIGs.

**Index Terms**—Domain-specific language, software language usability, human computer interaction, computer-interpretable guideline

## I. INTRODUCTION

Improving the quality of care depends on effective clinical decision support, as a recommendation based on best-practice guidelines can be provided at the point of care, as part of the clinician's workflow [1]. Computer-interpretable guidelines (CIGs) have been developed to integrate clinical practice guidelines (CPGs) into computer-supported clinical settings [2]. Furthermore, CIG modelling languages form the interface between CIG modellers and the CIGs they require to be executed in computer-based clinical decision support systems (CDSSs). But CPGs change frequently as the science and technology behind clinical practice improves, demanding changes in related CIGs as highlighted in previous studies [3], [4]. Such evolving CIGs require a CIG modelling language that can support computational management of such changes to ease the efforts required to maintain related CDSSs. Like other software languages, CIG modelling languages have rarely received usability or human factors evaluation which may have led to the deployment of inadequate languages [5].

This work evaluated the grammar of a novel CIG modelling language *FCIG* that enables computational management of CPG changes. The perception of novice and experienced CIG modellers towards the modelling language's usability was assessed. This work further highlights key language characteristics and related evaluation tools that were perceived to contribute to *FCIG*'s usability.

Section II gives a background on CPGs, CIGs and software language evaluation. Subsequently, Section III describes the language constructs that were evaluated. Section IV follows with the design of the evaluation study. The results are presented in Section V and discussed in Section VI. The paper is finally concluded in relation to future work in Section VII.

## II. BACKGROUND

This section provides some background on model-driven engineering and domain-specific languages. The section further continues with related work on CIG modelling and CPG changes.

### A. Clinical practice guidelines

Clinical Practice Guidelines (CPGs) are systematically developed statements to assist practitioner and patient decisions about appropriate healthcare for specific clinical circumstances [6]. CPGs describe the evidence-based procedures to be followed during clinical management of a specific health problem [2]. In addition, the CPGs aim to improve the quality of care, reduce unjustified variations, reduce costs and support introduction of new knowledge in clinical practice [7].

### B. Computer-interpretable guidelines

CIGs have been developed to integrate CPGs into computer-supported clinical settings [2]. A CIG is a representation of the knowledge that is required for a CDSS to advise clinicians in a way that adheres to CPGs [8]. There are two main approaches that have been used to express this kind knowledge. One approach is whereby the knowledge is inferred through case-based reasoning, chaining of individual rules, or individual decision rules [7]; e.g., Arden Syntax [9] and GELLO [10]. The other is whereby the clinical knowledge is modelled in a process-flow like model by representing knowledge in Task Network Models of clinical actions and decisions that unfold over time [7]; e.g., GLIF [11] and EON [12].

### C. Software language evaluation

A software language is any language that is created to describe and create software e.g. programming, modelling and query languages [13]. Software language users need to know the inherent strengths and domain applicability of the languages they use [14]. Quality goals of models coded in a particular software language can be categorised as syntactic, semantic and pragmatic referring to syntactic correctness, validity and completeness, and degree of comprehension respectively [15]. Evaluation criteria of software languages can also be classified into four categories [14]: 1) Application domain – also known as the philosophy of design [16] entails such criteria that assess how well a language supports developing software for a specific type of applications [17], [18]. Incompatible abstractions of the problem domain between

software language users and software language engineers are a constant challenge to language usability [19]; 2) Human factors – noting that software engineering is human-intensive, these criteria are used to assess user-friendliness of a software language [20]. Factors such as naturalness and readability of software languages have not been consistently endorsed by evidences in software language engineering [21]; 3) Software engineering – these are aspects that enable the development of good software by evaluating a language’s capacity to support such qualities as reliability, portability, reusability and maintainability [22]; 4) Language design and implementation – these assess how well a language is designed to enable support of modelling tools such as compilers and interpreters [23].

### III. THE SYNTAX OF *FCIG*

The subsections that follow describe *FCIG*’s language constructs using generalised and precise specifications with examples. For the precise specification, we use Xtext Extended Backus-Naur Form (EBNF) notation with its usual semantics as defined in [24]. For this generalised specification, we use the following notation and writing conventions. Let keywords be presented in bold and all optional keywords and user-defined terms be italicised. The notation uses three symbols to denote the three types of user-defined tokens for *FCIG*’s the grammar. The three symbols are described as follows:

- *NAME* should comprise of characters of any length, can contain letters, the underscore character and the digits zero to nine after the first initial letter.
- *STRING* should comprise of characters of any length, should be preceded and terminated by single or double quotes.
- *NUMBER* should comprise of digits, can include a period between any pair of digits to denote decimal numbers, and can be preceded by a dash to denote negative numbers.

*FCIG*’s grammar has three language constructs for specifying conditions, actions and guideline recommendations within a CPG. These three language constructs are discussed in the subsections that follow.

#### A. Conditions

To specify a Condition, the following syntax is adopted:

##### **Condition** [*NAME*]:

```
[Variable=STRING|NAME] [Relator=is | = | > | >=
| < | <=][Value=NUMBER|STRING|NAME]
true|false]
[Unit=STRING|NAME]
```

The precise concrete syntax using Xtext EBNF notation for the Condition construct is presented below:

```
Condition: 'Condition' name=ID ':'
decisionVariable=DecisionVariable
relator=Relator variableValue=Value
(unit=Unit)?;
```

The Condition construct begins with the keyword **Condition** followed by the name of the condition. A colon follows right after the name of the condition. After the colon, the decision variable, relator, variable value and optional unit for

the condition are specified in that order of sequence. For example, a Condition to depict a DNA-PCR test result that is positive can be written in *FCIG* as follows:

```
Condition positive_dna_result:
"DNA-PCR test result" is positive
```

Another example to specify a condition that depicts age of an infant as those that are under twelve months can be written in *FCIG* as follows:

```
Condition infant:
age < 12 months
```

#### B. Actions

To specify an Action, the following syntax is adopted:

##### **Action** [*NAME*]:

```
[Verb= STRING|NAME] [Verb Complement
= STRING|NAME]
```

The precise concrete syntax using Xtext EBNF notation for the Action construct is presented below:

```
Action: 'Action' name=ID ':'
verb=ActionVerb complement=VerbComplement;
```

The Action construct begins with the keyword **Action** followed by the name of the construct. A colon follows right after the name of the construct. Immediately following the colon, the Action Verb and Verb Complement for the Action construct are specified in that order of sequence. For example, an Action to prescribe a paediatric first-line regimen for ART can be written in *FCIG* as follows:

```
Action prescribe_paediatric_first_line_ART:
"Prescribe" "1P - D4T/3TC/NVP"
```

#### C. Recommendations

To specify a Recommendation, the following syntax is adopted:

##### **Recommendation** [*NAME*]:

```
Conditions [Condition], ...
Actions [Action], ...
```

The precise concrete syntax using Xtext [25] EBNF notation for the Recommendation construct is presented below:

```
Recommendation: 'Recommendation' name=ID ':'
'Conditions' conditions+=[Condition](','
conditions+=[Condition])*
'Actions' actions+=[Action](',' actions+=[Action])*;
```

The Recommendation construct begins with the Recommendation keyword followed by the name of the construct. A colon follows the name of the construct. After the colon, The Conditions keyword is specified and it is followed by comma separated names of Condition constructs. After the comma separated list of names of Condition constructs, the Actions keyword is specified and it is followed by a comma separated list of names of Action constructs. For example; a Recommendation for prescribing ART for infants that are not on ART, have a DNA-PCR test result that is positive and are aged under 12 months; that requires the first-line ART regimen to be prescribed for the patients; can be written as follows:

```
Recommendation ART_eligibility_upto_12_months:
Conditions positive_dna_result, infant,
not_on_ART
Actions prescribe_paediatric_first_line_ART
```

#### IV. MATERIALS AND METHODS

The aim of this study was to evaluate whether *FCIG*'s grammar is suitable and acceptable from a modeller's standpoint when modelling an evolving CIG. The user perceptions on the usability of *FCIG*'s language constructs were evaluated.

##### A. Research question

Recall, from the discussions in Section II, that incompatible abstractions between language users and language designers can pose challenges to usability of modelling languages [19]. In addition, perceived usability is an important high-level construct of usability [26], [27]. In order to focus on this perspective, the following research question were set:

**RQ1:** *Are the language constructs of FCIG perceived as usable?*

##### B. Research design

A mixed methods approach was adopted in the evaluation. A mixed methods approach combines quantitative and qualitative techniques, methods, approaches, concepts or language into a single study [28]. A mixed methods approach can provide strengths from one method that can make up for the weaknesses in another method, thereby developing rich insights into phenomena of interest that can not be fully understood using a single method [29]. The usability evaluation was separated into two smaller studies, one with novice CIG modellers and the other with experienced CIG modellers as study participants. Quantitative data was collected through a standardised usability questionnaire that used Likert scales. Qualitative data regarding grammar usability was further collected in order to understand the perception of CIG modellers when exposed to the language constructs that are provided in *FCIG*'s grammar.

##### C. Study setup

Study participants evaluated the language constructs of the grammar for *FCIG* modelling language. A paper-based orientation of the syntax of the grammar was given to all the study participants. For each study participant, the purpose of the CIG modelling language was explained to them. Thereafter, a scripted description of the three main language constructs of *FCIG* were given to the participants. Finally, the study participants evaluated the three main constructs of the grammar by responding to a usability questionnaire.

##### D. Study participants

Voluntary participation in the study was open to potential participants. Two categories of participants were recruited for the study through convenience sampling. The first category was that of novice CIG modellers with basic knowledge in computing but no experience in CDSS design and deployment recruited from the researchers' computer science postgraduate research laboratory. The second category was that of individuals with experience in CDSS design and deployment recruited from the researchers' network of CDSS designers and implementers. The study was planned to recruit at least five study participants in both categories of study participants.

##### E. Study design

A survey study was adopted for the evaluation. Survey studies are procedures in which investigators administer a survey to a sample or entire population to get a description of the attitudes, opinions, behaviours or characteristics of the population [30].

##### F. Data collection methods

Online questionnaires were used for data collection. The system usability scale (SUS) [31] that uses a five-point Likert scale was used as a standardised questionnaire to collect data about participants' perceptions on the usability of the grammar. Figure 1 shows a sample question from the questionnaire on such a Likert scale. The SUS was chosen as a standardised questionnaire because SUS is a simple and widely used ten-item scale that gives a global view of subjective assessments of usability [31]. The SUS as a standardised usability questionnaire underwent considerable psychometric evaluations which showed some evidence of its validity, reliability and sensitivity [26]. In addition, the SUS does not require a user licence [26], [27]. Two additional open-ended questions were further included in the online questionnaire in order to collect qualitative data that can give deeper insights where possible. Table I shows questions from number one to ten that were used for the SUS survey in addition to questions eleven and twelve for the qualitative survey.

TABLE I  
QUESTIONS FOR THE SURVEY

Question Number	Text
1	I think that I would like to use this grammar frequently
2	I found the grammar unnecessarily complex
3	I thought the grammar was easy to use
4	I think that I would need the support of a technical person to be able to use this grammar
5	I found the various functions in this language were well integrated
6	I thought there was too much inconsistency in this grammar
7	I would imagine that most people would learn to use this grammar very quickly
8	I found the grammar very cumbersome to use
9	I felt very confident using the grammar
10	I needed to learn a lot of things before I could get going with this grammar
11	Which keywords or concepts/functionality did you find useful whilst evaluating the grammar for FCIG?
12	Are there any features or keywords that are missing or need to be improved in the grammar?

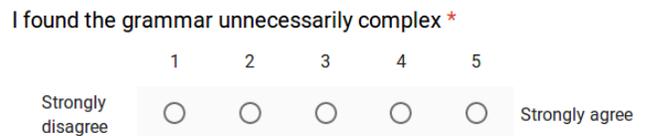


Fig. 1. Sample question with a five-point likert scale

##### G. Study protocol

For this study, each participant went through the following procedure:

- 1) The purpose of the evaluation was explained to the participant
- 2) The participant completed an informed consent form
- 3) The participant was given an average of a five minute description of the language constructs that make up *FCIG*'s grammar using suitable examples
- 4) The participant was given a link to complete a survey
- 5) The participant completed the survey

After following the study protocol, analyses and evaluations were carried out using the collected data.

#### H. Criteria to address the research question

In order to address research question from Section IV-A, SUS scores were calculated from the conducted surveys on a continuous scale that measures modellers' perceptions on language usability. Since *FCIG*'s language constructs are based on terminology that is commonly used in the clinical knowledge engineering domain, it was expected that CIG modellers would perceive the modelling grammar's usability with a more positive attitude. In order to guide the analyses, two alternative hypotheses were set.

The following hypothesis was formulated on the levels of SUS scores:

**H<sub>0</sub>:** CIG modellers do not perceive the language constructs of *FCIG* as usable.

**H<sub>1</sub>:** CIG modellers perceive the language constructs of *FCIG* as usable.

As the SUS surveys were conducted with two categories of CIG modellers, the following hypothesis was further formulated on the SUS score differences:

**H<sub>0</sub>:** There is no difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers.

**H<sub>1</sub>:** The SUS scores from experienced CIG modellers are higher than the SUS scores from novice modellers.

The *Shapiro-Wilk* test was used to test the samples of SUS scores from both experienced and novice CIG modellers for normality. In addition, the *Shapiro-Wilk* test helped in determining a relevant statistical test that helped in determining the statistical significance of observed differences between experienced and novice CIG modellers

## V. RESULTS

This section presents the results from the studies that were carried out to evaluate *FCIG* for its usability.

### A. Participants

There were six experienced CIG modellers and thirteen novice CIG modellers that participated in this study. Of the six experienced CIG modellers, one was recruited from South Africa whilst the rest were recruited from Malawi. The experienced modellers were university graduates with a computing background. In addition, all the experienced modellers worked for organisations that develop clinical decision support systems in low- and middle-income countries. All the novice CIG modellers were recruited from the Computer Science department at the University of Cape Town. Of the 13 CIG modellers, seven were students that were studying towards

a doctoral degree whilst the rest were studying towards a master's degree. The sections that follow present the results from the analysis that was carried out on the data that was collected through the SUS and qualitative surveys completed by the CIG modellers.

### B. Results from the SUS surveys

The sections that follow describe the results from the analysis of data that was collected from the usability study questionnaires.

1) *SUS score levels:* To start with, the first null and first alternative hypotheses were tested. Individual SUS scores were calculated from the surveys that were carried out with both experienced and novice CIG modellers. The sample of experienced CIG modellers had a mean of 89.17, a median of 91.25 and a mode of 90 for its SUS scores.

From the SUS scores that were calculated from the survey that was carried out with the novice CIG modellers, a mean of 79.23, a median of 80 and a mode of 80 were derived. Using the calculated SUS scores from both experienced and novice modellers, density plots were created that are presented in Figure 2 so the results could be visualised. Further to that, a box plot was also created for the resulting SUS scores that is presented in Figure 3. The density plots and box plot were created to carry out some exploratory analyses on the data that was collected through the questionnaire. The exploratory analyses revealed that the samples of SUS scores from both experienced and novice modellers might not have come from normally distributed populations. The analyses further revealed that the sample of SUS scores from experienced CIG modellers were higher than those that were computed from novice CIG modellers.

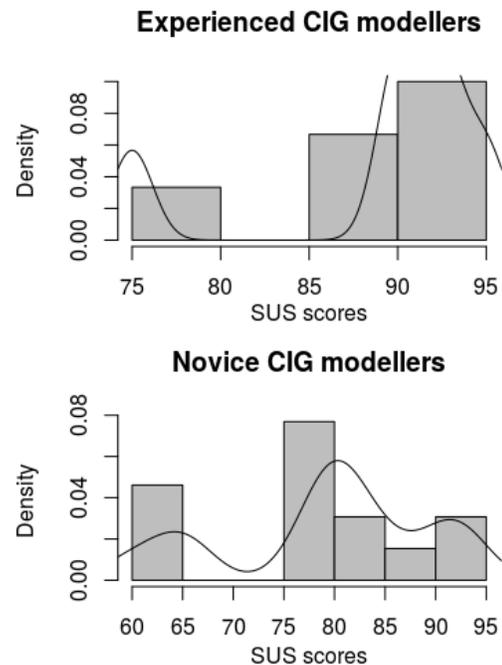


Fig. 2. Density plots of SUS scores on language constructs usability

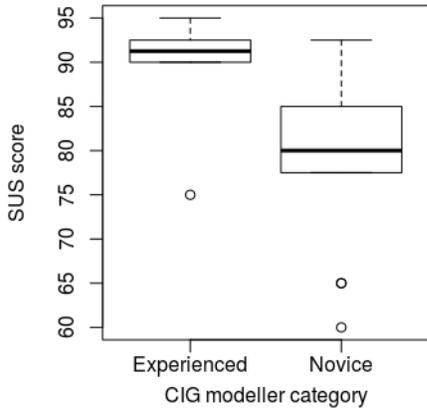


Fig. 3. Box plots of SUS scores on language constructs usability

TABLE II  
STATISTICS FOR SUS SCORES ON *FCIG* LANGUAGE CONSTRUCTS

CIG modeller level	SUS score		
	Mean	Median	Mode
Experienced	89.17	91.25	90
Novice	79.23	80	80

Thereafter, in-depth analyses of the collected data were carried out to test the hypotheses that were set out for this study. Recall the first null hypothesis,  $H_0$ : CIG modellers do not perceive the language constructs of *FCIG* as usable. Table II presents the statistics that were calculated from the samples of SUS scores on *FCIG* language constructs' usability. The mean and median values of the SUS scores from experienced and novice CIG modellers were both higher than 75. The results indicate that the first null hypothesis can be rejected. This means that there is evidence that CIG modellers perceived the language constructs of *FCIG* as usable. Therefore, we can say that both novice and experienced CIG modellers perceived *FCIG* as a usable CIG modelling language.

2) *SUS score levels between experienced and novice CIG modellers*: Thereafter, the second null and second alternative hypotheses were tested. The Shapiro-Wilk test was used to test the samples of computed SUS scores from the study for normality. Table III shows the details of the Shapiro-Wilk test results for all two samples which comprised of experienced and novice CIG modellers. The Shapiro-Wilk test result for the sample of SUS scores from experienced CIG modellers had a p-value that was small ( $p < 0.05$ ). Hence, the results showed that the SUS scores from the sample of experienced CIG modellers were not from a normally distributed population. Due to the fact that the SUS scores did not come from a normally distributed population, a Wilcoxon Rank Sum test was used to test for statistical significance of the differences measured from the two conditions of the study.

Recall the second null hypothesis,  $H_0$ : There is no difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers. The medians of SUS scores from experienced CIG modellers and

TABLE III  
SHAPIRO-WILK TEST RESULTS ON SUS SCORES OF LANGUAGE CONSTRUCTS USABILITY

CIG modeller category	W	p-value
Experienced	0.74	0.014
Novice	0.90	0.14

novice CIG modellers were 89.17 and 79.23, respectively. A Wilcoxon Rank Sum test was carried out to evaluate the differences in CIG modellers. The test showed that there was no significant effect of CIG modellers (The mean ranks of experienced CIG modellers and novice CIG modellers were 13.67 and 8.31, respectively;  $U = 0.90$ ,  $Z = 1.95$ ,  $p = 0.05$ ,  $r = 0.45$ ). Due to the fact that the effect of CIG modellers on SUS scoring was not significant, the second null hypothesis can not be rejected. This means that there is no significant difference in SUS scores between the ratings from experienced CIG modellers and those from novice CIG modellers.

## VI. DISCUSSION

Although SUS is a subjective assessment of usability, it can be considered as a valid and reliable indicator of usability [32]. It is argued that the SUS has undergone some considerable amount of psychometric testing to evaluate its validity, reliability and sensitivity [33], [26]. Furthermore, other studies have shown that results from the SUS can reliably converge at samples that are as low as eight [34], [26].

The results in Section V-A show that both novice and experienced CIG modellers rated the grammar of *FCIG* modelling language highly on the system usability scale. Hence it can be said that the participants in this study perceived the grammar of *FCIG* to be usable. Such a grammar evaluation can provide invaluable insights that can assist a software language designer to improve a software language of interest. Hence contributing to the likelihood of the software language to be adopted in practice.

From the ten dimensions of usability assessment on the SUS scale, the majority of both the novice and expert CIG modellers indicated that *FCIG*'s grammar was easy to use, had well integrated structural elements, did not have too much inconsistency in addition to not being very cumbersome to use. The reason for *FCIG* being perceived as a language with an easy-to-use and consistent grammar might stem from the fact that the language has a small and clear vocabulary consisting of just three main language constructs. Simplicity, clarity and consistency are widely accepted to be essential qualities of good modelling languages [35], [36], [37]. This can be further evidenced by the comments that the participants gave in their qualitative feedback after the survey. For instance, Participant *p5* commented that "*The three constructs explain the model well. Specifically Condition as it can load data that's vital to making an informed recommendation*".

Although the mean SUS scores for both novice and expert CIG modellers were different, the statistical test results in Section V-A show that the perception of experienced CIG

modellers on the usability of the grammar of *FCIG* was not significantly different to that of novice CIG modellers. This is an expected result because the grammar of *FCIG* uses a small set of language concepts that have a direct mapping to the clinical guideline formalization concepts. By employing a small set of language concepts with adequate expressive power, both novice and experienced modellers are likely to find the grammar as usable. Incompatible domain abstractions in a DSL grammar can introduce limitations that can negatively impact a DSL's usability [38], [19], [39].

## VII. CONCLUSION

*FCIG*'s grammar, a novel small and compact syntax for modelling CIGs, was evaluated for its usability in modelling evolving CIGs by both novice and experienced CIG modellers. Perceptions of CIG modellers were evaluated with regards to the grammar's usability. *FCIG* was perceived to have a pragmatic grammar for modelling CIGs. Both novice and experienced CIG modellers perceived *FCIG*'s grammar as highly usable and practical for modelling clinical practice guidelines. It is recommended that *FCIG* be further evaluated for its suitability and efficacy in modelling evolving CIGs by comparing it with other CIG modelling languages.

*Acknowledgements: This work was supported by the Hasso Plattner Institut through the Research Centre in ICT4D at the University of Cape Town. This conference presentation was supported by King's College London*

## REFERENCES

- [1] D. A. Handel and J. L. Hackman, "Implementing electronic health records in the emergency department," *Journal of Emergency Medicine*, vol. 38, no. 2, pp. 257–263, 2010.
- [2] A. González-Ferrer, A. ten Teije, J. Fernández-Olivares, and K. Milian, "Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks," *Artificial Intelligence in Medicine*, vol. 57, no. 2, pp. 91–109, 2013.
- [3] K. Kaiser and S. Miksch, "Versioning computer-interpretable guidelines: Semi-automatic modeling of 'living guidelines' using an information extraction method," *Artificial Intelligence in Medicine*, vol. 46, no. 1, pp. 55–66, 2009.
- [4] Y. J. Msosa, M. Densmore, and C. M. Keet, "Towards an architectural design of a guideline-driven EMR system: A contextual inquiry of Malawi," in *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*, ser. ICTD '15. ACM, 2015, pp. 49:1–49:4.
- [5] P. Gabriel, M. Goulao, and V. Amaral, "Do software languages engineers evaluate their languages?" *arXiv preprint arXiv:1109.6794*, 2011.
- [6] M. J. Field, K. N. Lohr, and others, *Guidelines for Clinical Practice:: From Development to Use*. National Academies Press, 1992.
- [7] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of Biomedical Informatics*, vol. 46, no. 4, pp. 744–763, 2013.
- [8] D. R. Sutton, P. Taylor, and K. Earle, "Evaluation of PROforma as a language for implementing medical guidelines in a practical context," *BMC Medical Informatics and Decision Making*, vol. 6, no. 1, p. 20, Apr. 2006.
- [9] G. Hripcsak, P. D. Clayton, A. T. Pryor, P. Haug, O. B. Wigertz, and J. Van der Lei, "The Arden Syntax for Medical Logic Modules," in *Proceedings. Symposium on Computer Applications in Medical Care*. American Medical Informatics Association, 1990, pp. 200–204.
- [10] M. Sordo, O. Ogunyemi, A. A. Boxwala, and R. A. Greenes, "GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support," *AMIA Annual Symposium Proceedings*, vol. 2003, p. 1012, 2003.
- [11] M. Peleg, A. A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado *et al.*, "Glif3: the evolution of a guideline representation format." p. 645, 2000.
- [12] S. W. Tu, M. A. Musen, and others, "Modeling data and knowledge in the EON guideline architecture," *Studies in health technology and informatics*, no. 1, pp. 280–284, 2001.
- [13] A. Kleppe, *Software language engineering: creating domain-specific languages using metamodels*. Pearson Education, 2008.
- [14] J. Howatt, "A Project-based Approach to Programming Language Evaluation," *SIGPLAN Not.*, vol. 30, no. 7, pp. 37–40, Jul. 1995.
- [15] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *IEEE software*, vol. 11, no. 2, pp. 42–49, 1994.
- [16] J. AlGhamdi and J. Urban, "Comparing and Assessing Programming Languages: Basis for a Qualitative Methodology," in *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing: States of the Art and Practice*, ser. SAC '93. New York, NY, USA: ACM, 1993, pp. 222–229.
- [17] M. Klerer, *Design of very high-level computer languages: a user oriented approach*. McGraw-Hill, Inc., 1991.
- [18] D. A. Watt, W. Findlay, and J. Hughes, *Programming language concepts and paradigms*. Prentice Hall Englewood Cliffs, 1990, vol. 234.
- [19] A. F. Blackwell, L. Church, and T. R. G. Green, "The abstract is' an enemy': Alternative perspectives to computational thinking," in *Proceedings PPIG*, vol. 8, 2008, pp. 34–43.
- [20] C. Ghezzi and M. Jazayeri, *Programming language concepts*. New York: Wiley, 1997.
- [21] S. Markstrum, "Staking claims: a history of programming language design claims and evidence: a positional work in progress," in *Evaluation and Usability of Programming Languages and Tools*. ACM, 2010, p. 7.
- [22] M. Shaw, G. T. Almes, J. M. Newcomer, B. K. Reid, and W. A. Wulf, "A comparison of programming languages for software engineering," *Software: Practice and Experience*, vol. 11, no. 1, pp. 1–52, 1981.
- [23] N. Wirth, "On the Design of Programming Languages," in *IFIP Congress*, vol. 74, 1974, pp. 386–393.
- [24] L. Bettini, *Implementing Domain-Specific Languages with Xtext and Xtend*. Packt Publishing Ltd, 2013.
- [25] H. Behrens, M. Clay, S. Efftinge, M. Eysholdt, P. Frieze, J. Köhnlein, K. Wannheden, and S. Zarnekow, "Xtext user guide," 2008. [Online]. Available: [https://eclipse.org/Xtext/documentation/1\\_0\\_1/xtext.pdf](https://eclipse.org/Xtext/documentation/1_0_1/xtext.pdf)
- [26] J. Sauro and J. R. Lewis, *Quantifying the user experience: Practical statistics for user research*. Elsevier, 2012.
- [27] J. Brooke, "SUS: A Retrospective," *J. Usability Studies*, vol. 8, no. 2, pp. 29–40, Feb. 2013.
- [28] B. R. Johnson and A. J. Onwuegbuzie, "Mixed methods research: A research paradigm whose time has come," *Educational researcher*, vol. 33, no. 7, pp. 14–26, 2004.
- [29] V. Venkatesh, S. A. Brown, and H. Bala, "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems," *MIS quarterly*, vol. 37, no. 1, 2013.
- [30] J. W. Creswell, *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Pearson, 2011.
- [31] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.
- [32] J. R. Lewis and J. Sauro, "The factor structure of the system usability scale," in *Human centered design*. Springer, 2009, pp. 94–103.
- [33] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *J. Usability Studies*, vol. 4, no. 3, pp. 114–123, May 2009.
- [34] T. S. Tullis and J. N. Stetson, "A comparison of questionnaires for assessing website usability," in *Usability professional association conference*, 2004, pp. 1–12.
- [35] T. W. Pratt, M. V. Zelkowitz, and T. V. Gopal, *Programming languages: design and implementation*, 1984.
- [36] R. F. Paige, J. S. Ostroff, and P. J. Brooke, "Principles for modeling language design," *Information and Software Technology*, vol. 42, no. 10, pp. 665–675, 2000.
- [37] G. Karsai, H. Krahn, C. Pinkernell, B. Rumpe, M. Schindler, and S. Völkel, "Design guidelines for domain specific languages," in *9th OOPSLA Workshop on Domain-Specific Modeling (DSM09)*, 2009, pp. 7–13.
- [38] M. Mernik, J. Heering, and A. M. Sloane, "When and How to Develop Domain-specific Languages," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, Dec. 2005.
- [39] A. F. Blackwell and C. Morrison, "A logical mind, not a programming mind: Psychology of a professional end-user," in *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group (PPIG 2010)*, 2010, pp. 19–22.