



Towards a GUI for Declarative Medical Image Analysis: Cognitive and Memory Load Issues

Giovanna Broccia^(✉), Vincenzo Ciancia, Diego Latella, and Mieke Massink

Consiglio Nazionale delle Ricerche, Istituto di Scienze e Tecnologie dell'Informazione
'A. Faedo', Pisa, Italy

{giovanna.broccia,vincenzo.ciancia,diego.latella,
mieke.massink}@isti.cnr.it

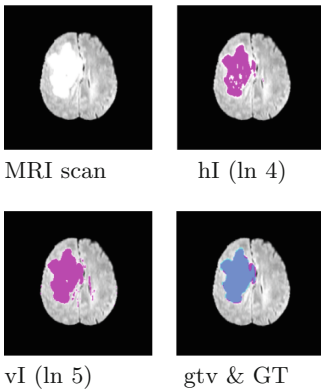
Abstract. In medical imaging, (semi-)automatic image analysis techniques have been proposed to support the current time-consuming and cognitively demanding practice of manual segmentation of *regions of interest* (ROIs). The recently proposed *image query language* ImgQL, based on spatial logic and model checking, represents segmentation methods as concise, domain-oriented, human-readable procedures aimed at domain experts rather than technologists, and has been validated in several case studies. Such efforts are directed towards a *human-centred* Artificial Intelligence methodology. To this aim, we complemented the ongoing research line with a study of the Human-Computer Interaction aspects. In this work we investigate the design of a graphical user interface (GUI) prototype that supports the analysis procedure with minimal impact on the focus and the memory load of domain experts.

Keywords: User-centred design · Usability study · Cognitive evaluation · Medical image analysis · Spatial logic · Spatial model checking

1 Introduction

In the field of Medical Imaging [17], *segmentation* plays a crucial role [14]. Segmentation consist in the identification of regions of interest (ROIs) – that might correspond to tissue, organ, traces of disease, or other relevant structure – and dividing the image into meaningful segments. The current practice of manual segmentation (e.g., in Radiotherapy) is time-consuming and cognitively demanding, justifying the plethora of existing research on fully- and semi- automatic segmentation methods. In recent work (see [1–3, 9]), the approach of *spatial model checking* has been proposed. Such methodology combines local image features

This work has been partially supported by the Italian MIUR-PRIN 2017 project IT-MaTTerS: Methods and Tools for Trustworthy Smart Systems.



Specification 1: Tumour segmentation

```

1 // Normalisation
2 let pflair = percentiles(flair,brain,0)
3 // High and low thresholds
4 let hI = pflair > . 0.95
5 let vI = pflair > . 0.86
6 // Remove noise
7 let hyperIntense = flt(5.0,hI)
8 let veryIntense = flt(2.0,vI)
9 //Semantic noise removal
10 let growTum = grow(hyperIntense, veryIntense)
11 // Statistical texture similarity
12 let tumSim = similarFLAIRTo(growTum)
13 let tumStatCC = smoothen(2.0,tumSim > . 0.6)
14 // Tumour segmentation
15 let gtv = grow(growTum,tumStatCC)
16 save "hyperIntense.nii.gz" hyperIntense
17 print "00_dice_gtv" diceM(gtv)

```

Fig. 1. ImgQL segmentation of image Brats17_2013.7_1: Overlays hI, vI and gtv (pink) and ground truth (GT) (blue). (Color figure online)

(intensity, colour, texture) with spatial/topological characteristics (relative distance, contact, connectedness), exploiting a domain-specific, concise, declarative language named ImgQL (“image query language”) and the efficient, parallel spatial model checker *VoxLogicA*¹, optimised for operating on medical images, e.g. MRI or CT scans (see [10] for a tutorial). The resulting ROIs are meant to be visualised as (semi-transparent) number-valued as well as boolean (so-called “masks”) layers on top of the original images. Common similarity indexes, such as Dice-Sorensen, can be directly defined in ImgQL (see e.g. Fig. 1).

VoxLogicA has been successfully applied to BraTS 2017², a publicly available set of benchmark MRI images for brain tumour segmentation [3]. The obtained results are well in line with the state of the art, both in terms of accuracy and in terms of computational efficiency, while supporting explainability, easy replicability and exchange of analysis methods. So far, the approach has been used via a command line interface. This is sufficient for the technical validation of the tool, but hinders usability and broader uptake by domain experts.

The GUI design proposed in the present paper follows a user-centred approach, taking into consideration typical tasks performed by different classes of professionals in this domain [13]. In particular it aims at supporting the analysis procedures with minimal impact on the focus of attention and on the user’s memory load [5, 6]. The *handling of layers* is identified as one of the central tasks. The creation and visualisation of layers not only supports ROI identification but also the understanding and evaluation of the results of ImgQL queries. We analyse the effect of layers selection from the GUI on the capability of users to perform common tasks (visualise the layers identified by a logic specification, identify a new layer via a logic formula, visualise and analyse a new layer identified by a

¹ *VoxLogicA* is available at <https://github.com/vincenzoml/VoxLogicA>.

² See <https://www.med.upenn.edu/sbia/brats2017/data.html>.

logic formula, compare two different layers) in terms of time performance, error in the execution of tasks, and memory load, using user tests and by a theoretical evaluation of the memory load. We focus on the following research questions:

I RQ1. *What is the effect of layers selection on task performance?*

II RQ2. *Is the relation between the layers and the specification clear to users?*

III RQ3. *How does GUI vs. command line layer selection affect memory load?*

To answer RQ1 and RQ2, we performed a usability study that provided useful feedback on the efficiency and effectiveness of the layers handling mechanism. To answer RQ3 we performed a theoretical evaluation on the memory load required to complete the tasks conducted in the usability test, with and without the support of the GUI. In general, the design of GUIs for medical imaging tools is mainly inherited from the medical consoles already used in the medical domain, without a dedicated study on their usability [8]. In [12] different user interfaces for 2D and 3D imaging are reviewed and discussed, and the usability of such tools is regarded as essential for their adoption by clinicians. In [11, 15] a number of tools with GUI have been evaluated regarding a set of aspects included usability, that has been assessed qualitatively as the result of the authors' experience.

For the design of the GUI three classes of users were considered: 1) *Healthcare professionals*. This class consists of clinicians that generally use image segmentation. Typical tasks are manually inspecting and comparing base images and related ROIs; 2) *Researchers in Medical Imaging*. Their goal is to find innovative solutions to support the work of clinicians; 3) *IT professionals*. These are developers that want to create and propose tools and techniques to support the work of clinicians. Independently from the user class, image segmentation tasks are cognitively demanding, especially from a memory point of view, due to the need to manage a variety of information of very different nature such as case information, medical records, dictation systems, reporting facilities, literature search, access to databases and version control, to mention a few [18].

2 A GUI for Declarative Medical Image Analysis

Figure 2 shows the underlying architecture of VoxLogicA and its GUI. From the analysis of different tasks related to different classes of users, we conjecture that the GUI displaying both input and output in a single window plays a central

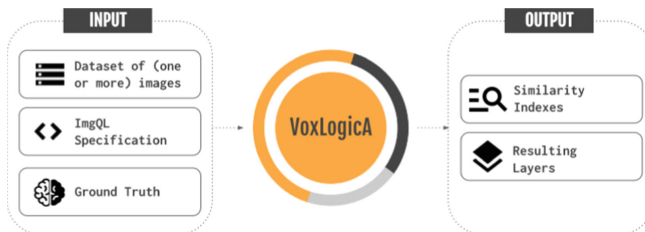


Fig. 2. The VoxLogicA process and architecture

role in reducing the user’s memory load: having all the necessary information on display helps users to complete their tasks without having to retrieve them from their working memory at each step.

The GUI prototype has been implemented using HTML, CSS and JavaScript and it runs as a desktop application through Electron³. The next version of the GUI is under development, building upon the research that we present in this paper. No longer being a prototype, the next version will be based on a state-of-the-art web user interface framework, namely `vue.js` version 3, using `pinia` as its store⁴.

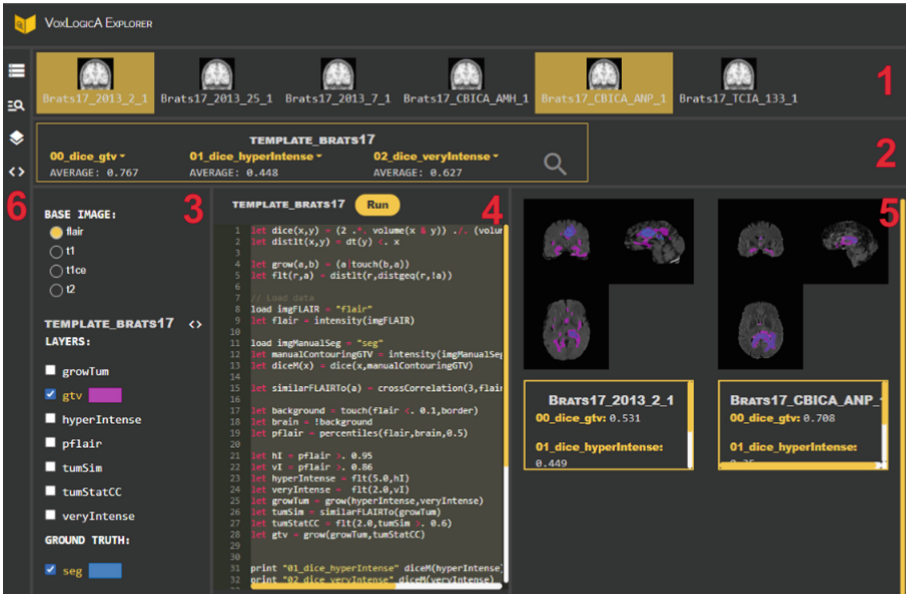


Fig. 3. VoxLogicA graphical user interface

Figure 3 shows the complete GUI prototype with all elements open (the red numbers in the figure are for reference only and are not part of the GUI). Below we describe its elements: the name of the GUI element and the corresponding number introduce each item.

Dataset Row (1). This section shows the dataset of images. To analyse one or more images more in detail, users can open/close cases by clicking on the thumbnail images. This action changes the thumbnail images background colour to yellow and opens/closes an embedded DICOM viewer for each image in the *work space* presented below.

³ Electron (<https://www.electronjs.org/>) is an open-source software framework which allows for the development of desktop GUI applications using web technologies.

⁴ See <https://vuejs.org/> and <https://pinia.vuejs.org/>.

Indexes Row (2). A section immediately below the *dataset row* shows the similarity indexes computed over that dataset by the `VoxLogicA` specification shown in the *code column*. Moreover, in this section one can look for items in the *dataset row* that satisfy specific characteristics, using a dedicated search box that opens when clicking on the search icon.

Layers Column (3). The bottom-left column of the window shows the list of base images and the list of overlays saved with the `VoxLogicA` specification shown in the *code column*. The base image is activated by clicking on the related button: when selecting a case from the dataset row, the system will display the activated base image in the viewer referring to the chosen case in the work space. Moreover, by clicking one or more overlays they will be opened as transparent layers on top of the base image in the work space. The system automatically selects a different colour for each layer. If present, the ground truth can be shown as a layer. Its presence is indicated after the title “GROUND TRUTH”.

Code Column (4). In the bottom-middle column of the window the `ImgQL` specification is shown through an embedded code editor. It can be edited and the modified specification can be run on the open cases shown in the workspace by clicking the button “Run”; if no cases are open, the button is not active.

Work Space (5). In the bottom-right column of the window the open cases are displayed (instantiated with the active base image), together with the active overlays, through embedded image viewers. Below each image viewer a box is shown with the information about the open case: its name and the values for the similarity indexes computed by the specification.

Icons Column (6). A column on the left side of the window shows four icons by clicking on which one can show or hide associated sections in the window.

3 GUI Evaluation

We are performing a number of usability studies on different classes of users to test specific GUI features. Each of these studies is conducted on small groups of users, following an iterative design cycle. Here we present a usability study performed on 5 users belonging to the academic research area (the group is composed of computer science students and professors recruited from a master class on spatial logic and model checking) focusing on the evaluation of the handling of the layers, in order to answer RQ1 and RQ2 (see Sect. 1). Moreover, to answer RQ3, we performed a theoretical evaluation on the memory load required to complete the tasks proposed in the usability study, comparing two variants of these tasks, one performed with the support of the GUI and one performed via command line operations.

3.1 Usability Study

The study design reproduces a realistic scenario, where users were asked to evaluate the quality of a `VoxLogicA` segmentation procedure with the support

of the GUI. Users were asked to perform selected representative tasks and to fill out a post-study questionnaire under the supervision of a test moderator who guided and observed the participants during the test execution. The study started with a familiarisation phase during which participants freely used the **VoxLogicA** GUI. The central phase of the study consisted of the completion of 4 tasks, each of which composed of a variable number of basic tasks. For each basic task users provided an answer or performed an action on the GUI which were annotated by the test moderator. Each task has been developed to test a specific GUI feature. Here we focus on the following tasks:

1. Visualise and analyse results regarding the images identified in previous tasks
2. Modify the segmentation procedure by adding and saving a new layer and compare the new layer with a previous version

At the end of the central phase participants were asked to fill out a post-study questionnaire composed of 3 sections: one on personal information, one on feedback and suggestions about the GUI and one on GUI satisfaction.

Results. With the data collected through the post-study questionnaire, feedback was gathered on the GUI, as well as on its desired new features. Here we present only those aspects concerning the handling of layers.

Participants were asked to rate the comprehensibility of the opening and closing of layers on a 5-points scale ranging from 1 (“hardly comprehensible”) to 5 (“very comprehensible”). The collected answers attest that this aspect is very comprehensible (4 participants rated 5 and 1 participant rated 4). Users were also asked how clear it was for them that the layers visualised in the layers column are all and only the ones computed and saved with the **VoxLogicA** procedure exposed in the code column. The collected answers attest that this is less well-understood: 1 user rated 2 in the 5-points scale, while the remaining users rated in equal measure 4 and 5.

Considering the users’ perception on whether the GUI features are able to support them in completing the usability study they all responded that they are satisfied with the GUI features, except for one user who would find it useful to have additional support for understanding the **VoxLogicA** specification (such as a *reading guide* or a description of the **VoxLogicA** primitive operators). Regarding the collected suggestions on how to improve the GUI, we received a number of helpful proposals. Suggested were the availability of an initial guide where the basic aspects of the interface are explained and the improvement of the automatic choice of colours when multiple layers have to be displayed together (e.g. using colours with high contrast to help the visualisation, or letting the users choose the colours they prefer).

The data gathered by the test moderator on the users’ performance, provides some quantitative measures on the central phase of the usability test. Here we

only report the task success rate (namely the percentage of *successfully* completed tasks)⁵ [16]. Although all tasks have been completed by all users, in 4 cases users had some uncertainties in completing (part of) a task, resulting in a success rate of 80%. The main issues we observed regard task 2: some users experienced difficulties in (a) adding the new layer and (b) comparing the new and old layers. Difficulties in case (a) mainly concerned the comprehension of the `VoxLogicA` language (ImgQL). This is also confirmed by one of the suggestions collected during the post-study questionnaire. While difficulties in case (b) concern the choice of colours automatically performed by the system. This is again confirmed by another suggestion collected during the questionnaire.

User satisfaction and perceived usability of the GUI was measured via a System Usability Scale (SUS) questionnaire provided at the end of the test [7]. Overall, based on this questionnaire, the prototype GUI was perceived as sufficiently usable and satisfying, with a mean score of 82.5. Just one participant gave a global score of 62.5, below sufficient evaluation.

3.2 Memory Load Evaluation

To answer RQ3 (see Sect. 1), we performed a theoretical evaluation of the amount of information needed by the users to complete the tasks proposed in the usability test with and without the support of the GUI (details in [4]).

When the tasks are performed without the support of the GUI users have to navigate the file system in order to find input and output images they need. This implies remembering at each step not only the names of the images but also the names of the directories, their position and whether they have been already opened or not. To this, we have to add all the information regarding the use of additional tools for open and visualise MRI scans, analysing the similarity indexes, updating the specification code and running the analysis through command line operations. With the support of the GUI, instead, all the required information is displayed, available and visible. Users do not need to navigate the file system (and thus remember names and positions) since they already have both input and output files available on the GUI. They do not need to remember how to use external tools since the GUI provides both an embedded code editor and viewers that enable a simplified task execution. Finally, users do not need to remember how to run `VoxLogicA` from command line since they can perform an analysis by just clicking on a button. This results in a considerable reduction of the memory load in using the proposed GUI.

4 Conclusion and Future Work

We presented a prototype of a novel GUI for various user classes in the domain of medical image segmentation. The GUI supports the use of the spatial model

⁵ The success rate is measured as the sum of the tasks completed successfully and the tasks completed with some issue weighted 0.5, over the total number of completion attempts (i.e. the product of the number of tasks and the number of participants).

checker **VoxLogicA** for the design of novel medical image analysis methods. We evaluated a first GUI prototype with a usability study and a memory load evaluation. The study provides important feedback on how to improve the GUI for further analysis. The memory evaluation shows how the GUI can help users in completing their tasks successfully with a lower cognitive effort. As part of future work, we plan to carry on the usability studies on other classes of users to test further GUI features, in order to improve the tool in an iterative design cycle.

References

1. Banci Buonamici, F., Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Spatial logics and model checking for medical imaging. *Int. J. Softw. Tools Technol. Transfer* **22**(2), 195–217 (2019). <https://doi.org/10.1007/s10009-019-00511-9>
2. Belmonte, G., Broccia, G., Ciancia, V., Latella, D., Massink, M.: Feasibility of spatial model checking for nevus segmentation. In: *FormaliSE*, pp. 1–12 (2021)
3. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: **VoxLogicA**: a spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) *TACAS 2019*. LNCS, vol. 11427, pp. 281–298. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17462-0_16
4. Broccia, G., Ciancia, V., Latella, D., Massink, M.: A graphical user interface for medical image analysis with declarative spatial logic - cognitive and memory load evaluation. Technical report, ISTI Technical report, ISTI-2021-TR/012, pp. 1–39 (2021)
5. Broccia, G., Masci, P., Milazzo, P.: Modeling and analysis of human memory load in multitasking scenarios: late-breaking results. In: *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pp. 1–7 (2018)
6. Broccia, G., Milazzo, P., Ólveczky, P.C.: Formal modeling and analysis of safety-critical human multitasking. *Innov. Syst. Softw. Eng.* **15**(3), 169–190 (2019). <https://doi.org/10.1007/s11334-019-00333-7>
7. Brooke, J.: SUS: a quick and dirty’ usability. *Usab. Eval. Ind.* **189**(3) (1996)
8. Cannella, V., Gambino, O., Pirrone, R., Vitabile, S.: GUI usability in medical imaging. In: *2009 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 778–782. IEEE (2009)
9. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Model checking spatial logics for closure spaces. *Log. Methods Comput. Sci.* **12**(4) (2016). <http://lmcs.episciences.org/2067>
10. Ciancia, V., Belmonte, G., Latella, D., Massink, M.: A hands-on introduction to spatial model checking using **VoxLogicA**. In: Laarman, A., Sokolova, A. (eds.) *SPIN 2021*. LNCS, vol. 12864, pp. 22–41. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84629-9_2
11. Haak, D., Page, C.E., Deserno, T.M.: A survey of DICOM viewer software to integrate clinical research and medical imaging. *J. Dig. Imag.* **29**(2), 206–215 (2016)
12. Iannessi, A., Marcy, P.-Y., Clatz, O., Bertrand, A.-S., Sugimoto, M.: A review of existing and potential computer user interfaces for modern radiology. *Insights Imaging* **9**(4), 599–609 (2018). <https://doi.org/10.1007/s13244-018-0620-7>
13. Ergonomics of human-system interaction - Human-centred design for interactive systems. Standard, International Organization for Standardization (2019). <https://www.iso.org/standard/77520.html>

14. Koundal, D., Kadyan, V., Dutta, P., Anand, V., Aggarwal, S., Gupta, S.: Computational techniques in biomedical image analysis: overview. In: *Advances in Computational Techniques for Biomedical Image Analysis*, pp. 3–31 (2020)
15. Liao, W., Deserno, T.M., Spitzer, K.: Evaluation of free non-diagnostic dicom software tools. In: *Medical Imaging 2008: PACS and Imaging Informatics*, vol. 6919, pp. 11–22. SPIE (2008)
16. Nielsen, J., Budiu, R.: Success rate: the simplest usability metric (2001). <https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/>
17. Ritter, F., Boskamp, T., Homeyer, A., Laue, H., Schwier, M., Link, F., Peitgen, H.O.: Medical image analysis. *IEEE Pulse* **2**(6), 60–70 (2011)
18. Sharma, A., Wang, K., Siegel, E.: Radiologist digital workspace use and preference: a survey-based study. *J. Digit. Imaging* **30**(6), 687–694 (2017)