Informatics for Health & Social Care
January 2010; 35(1): 10–24

informa
healthcare

Measuring the impact of the computer on the consultation: An open source application to combine multiple observational outputs

BERNHAR PFLUG[1], PUSHPA KUMARAPELI[2], JEREMY VAN VLYMEN[2], ELSKE AMMENWERTH[1] & SIMON DE LUSIGNAN[2]

[1]*Department of Medical Informatics and Technology, University of Health Sciences, Tyrol, Austria and* [2]*Division of Community Health Sciences, Department of Primary Care Informatics, St. George's University of London, London, UK*

Abstract
A diverse range of tools and techniques can be used to observe the clinical consultation and the use of information technology. These technologies range from transcripts; to video observation with one or more cameras; to voice and pattern recognition applications. Currently, these have to be observed separately and there is limited capacity to combine them. Consequently, when multiple methods are used to analyse the consultation a significant proportion of time is spent linking events in one log file (e.g. mouse movements and keyboard use when prescribing alerts appear) with what was happening in the consultation at that time. The objective of this study was to develop an application capable of combining and comparing activity log-files and with facilities to view simultaneously all data relating to any time point or activity. Interviews, observations and design prototypes were used to develop a specification. Class diagram of the application design was used to make further development decisions. The application development used object-orientated design principles. We used open source tools; Java as the programming language and JDeveloper^{TM} as the development environment. The final output is log file aggregation (LFA) tool which forms part of the wider aggregation of log files for analysis (ALFA) open source toolkit (www.biomedicalinformatics.info/alfa/). Testing was done using sample log files and reviewed the application's utility for analysis of the consultation activities. Separation of the presentation and functionality in the design stage enabled us to develop a modular and extensible application. The application is capable of converting and aggregating several log files of different formats and displays them in different presentation layouts. We used the Java Media Framework to aggregate video channels. Java extensible mark-up language (XML) package facilitated the conversion of aggregated output into XML format. Analysts can now move easily between observation tools and find all the data related to an activity. The LFA application makes new analysis tasks feasible and established tasks much more efficient. Researchers can now store multiple log file data as a single file isolate and investigate different doctor–computer–patient interaction.

Keywords: *Video recordings, process assessment, computer, general practice, medical records system, computerised, family practice, clinical consultation, software, open source*

Correspondence: S. de Lusignan, Division of Community Health Sciences, Department of Primary Care Informatics, St. George's University of London, London, SW17 0RE, UK. Tel: +20-8725-5661. E-mail: slusigna@sgul.ac.uk

ISSN 1753-8157 print/ISSN 1753-8165 online © 2010 Informa UK Ltd.
DOI: 10.3109/17538150903358693

1. Introduction

Although UK primary care is largely delivered through computer-mediated consultations [1], there are few established methods to compare the impact of different brands of computer systems on the consultation. The computer has improved the efficiency of consultation tasks [2,3] and has become an increasingly integral component of the consultation. General practitioners use a range of different clinical computer systems that were largely developed in isolation from each other. Regardless of the important role played by the computers in primary care, the impact they have on the consultation tasks and their relationships between the multitudes of computer system design features remains poorly investigated. Widely used assessment techniques are primarily based on manual observations [4,5] possibly they were developed in a way that minimises processing load. Consultation rating scales are usually quantitative and the outputs usually consist of a series of duration variables [6]. Automated observation techniques using activity recording tools [7] or image pattern recognition methods [8] also create activity log files with time stamps or duration variables of different formats.

Evaluation of the clinical consultation is hard because there are a lack of specific tools which allow objective assessment of: (1) Consultation interactions; (2) Take account of variations of output formats from the available methods of consultation monitoring; (3) Difficulty in combining multiple visual and quantitative measurements; (4) The challenge of getting a unified overview of the whole consultation and (5) How time consuming it is to find the same task in multiple consultations (e.g. How General Practitioners (GP) respond to prescribing alerts).

There are existing tools that allow combining of multiple observational data streams, however these are not capable of accepting outputs from heterogeneous origins. Qualitative research tools like Atlas. Ti™ and QSR NVIVO™ are capable of combining text and multimedia data sources. However, they cannot accept direct outputs from external sources like automated computer use monitoring applications. Usability testing applications such as Morae™ do not produce combined outputs that are adjustable according to research needs and are only capable of accepting only fixed number of observational inputs. Other commonly used observational data analysis tools like GameBreaker™ or Transana™ are not capable of producing comparable outputs from a series of consultations or classifying consultation interactions in a meaningful way [9].

In the absence of any suitable application to process observational data covering multiple aspects of doctor–computer–patient interactions, we decided to develop a data aggregation tool that would enable researchers to combine observations from diverse origins into a single analysable output. We decided to call this tool the log file aggregation (LFA) tool.

2. Method

2.1. LFA toolkit specification development

We reviewed the existing computer-mediated consultation observation and analysis techniques to develop a system specification for the aggregation tool. We used a combination of a literature review, observations and interviews as our requirement capturing techniques. First we reviewed the existing publications to identify the aims of the research method, its utility, limitations and potential areas for improvements. We then carried out unstructured interviews with the researchers who have been using the manual method to analyse consultation; this involved manual coding of interactions using an observational data capture (ODC) tool (Figure 1). Initially we used a proprietary tool called ObsWin™ though more recently we (led by PK) have developed our own open source ODC tool to overcome problems with screen size and the inability of ObsWin to allow 'rewind' if you miss an item you are coding.

We also interviewed those who were involved in developing the new application to explore their understanding of current problems and potential solutions. Further investigations included: (1) Examining the log files generated by each technique, (2) Observing researchers using think-aloud during data analysis and (3) Studying variations of result presentation formats.

We decided to make any output from our work open source and available under a GNU General Public Licence (GPL). We opted for this because of its 'copyleft' principles [10]. Copyleft allows developers to copy and adapt the source code, however these derived works must remain available under the same principles. This ensures that added to or derived works are still available to users. This licence makes this and other components of the aggregation of log files for analysis (ALFA) toolkit freely available to researchers and for non-commercial use from http://www.biomedicalinformatics.info/alfa/ conditional that they reference an earlier paper describing the tool kit's development is referenced [9].

Figure 1. The ODC used to manually code activities in the consultation of interest. (The parts shown are the main OCD menu and the small on-screen rating interface).

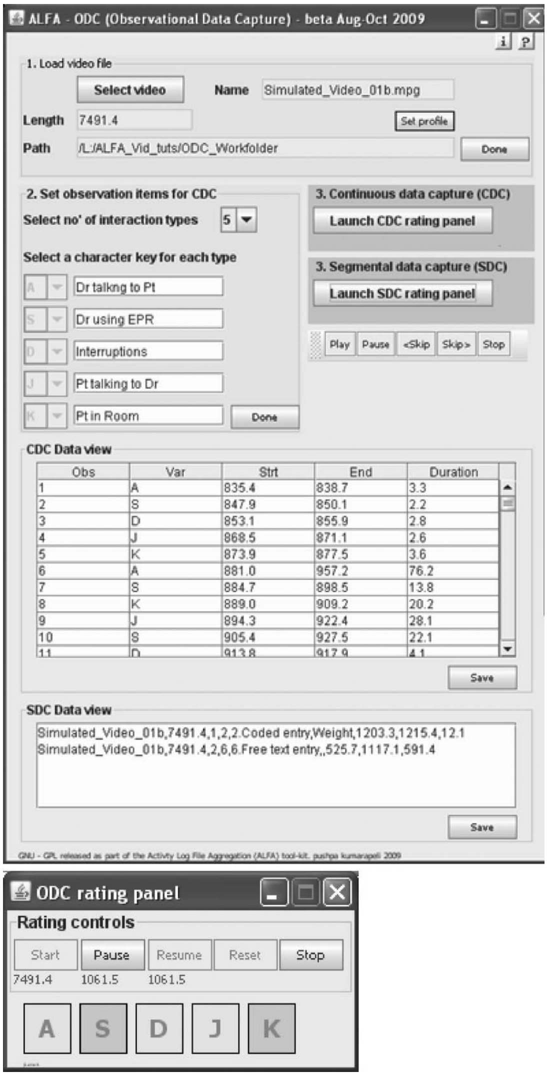


Figure 1. (*Continued*).

2.2. Modelling

We used Unified Modelling Language (UML) [10] to model then subsequently develop and describe the design of the LFA tool. We utilised the UML Class diagram to graphically represent the rational entities of the design, its architecture, interfaces and modules. Rational Unified Process [11] was selected as the software development process. This involved iterative and incremental development of the software combined with feed back sessions. The UML class diagram was used as the main point of reference for each design decision.

We implemented the UML model of the aggregation tool as an Object Oriented Application (OOA). We used JDeveloper™ a free integrated software development environment (from Oracle). JDeveloper provides a supports programming, editing, debugging and interface design. The programming was done using Java Standard Edition (J2SE). We promoted reusability and modularity of components whenever possible. We referred to the process specification to define the boundaries for the component modules. Particular attention was given to the separation of the application logic, (data processing) from the presentation logic (data display) as this would benefit any potential extensions to the application [12].

2.3. Developing the data processing and data display components

After meeting the functional requirements (i.e. essential functionality) through the system development, we further reviewed the user expectations to identify enviable features. This aimed at identifying the potential to allow the software to be compatible with other observation techniques. We also aimed at enhancing the data presentation format by the addition of more customisable features for each processing stage.

2.4. Testing

We tested the software to assure its capability to meet the system requirements. Our testing stage included checks for correctness, completeness, reliability, validity and usability. The first four test categories aimed at identifying the faults of software design and its logic to meet the requirement specification. Usability testing primarily focused on the interface aspects of the software to assess the user experience. Accurate reading, writing and displaying was tested using sample log-files. We designed a pilot study comparable to a previously done research using manual processing steps.

2.5. Ethical considerations

We developed this technique using the output from simulated consultation. No real consultation data were used during the development and testing stages, or visible in the filming locations. However, this application has subsequently been used for the analysis of routine consultations, as part of a study approved by a research ethics committee.

3. Results

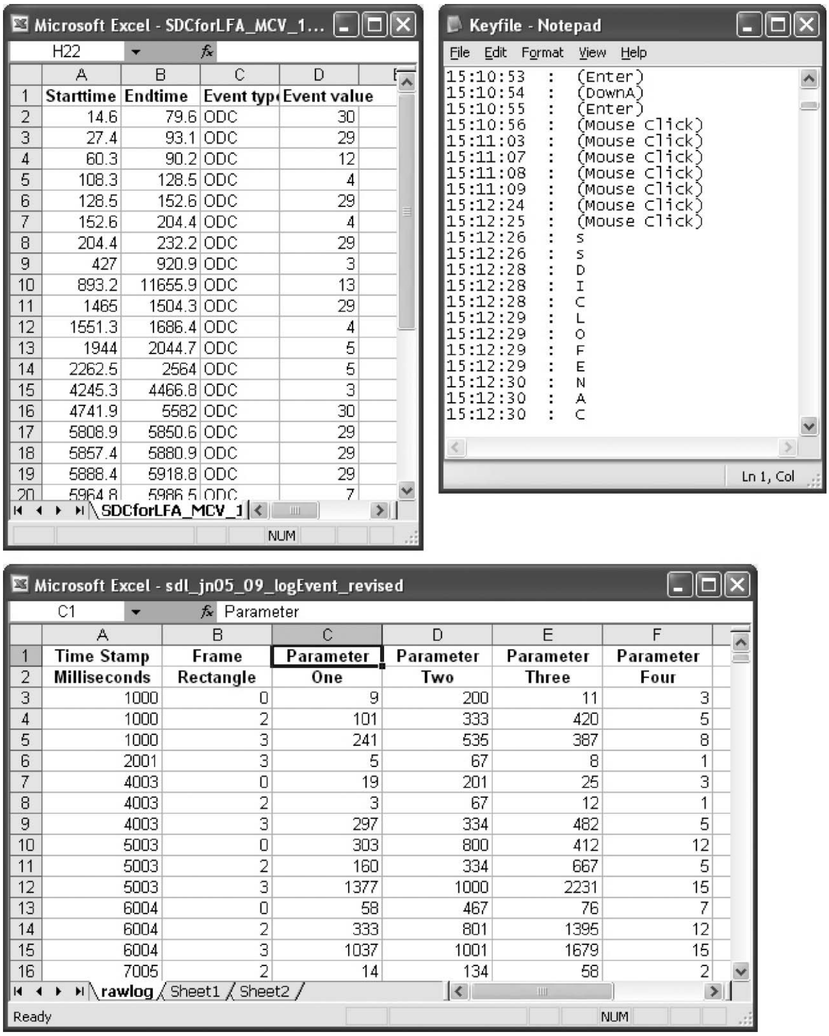
3.1. Specification for the LFA tool

The core components of the specification we developed were: (1) Need to be able to incorporate any number of log files; (2) The ability to link these to a media file – in our work multi-channel video; (3) The final LFA tool had to provide a single file of all the data other than the media file and provide an overview of the consultation and (4) We needed a common output format which allowed summary data to be used in a variety of ways.

Our LFA tool needs to be able to accept log file data from a wide range of sources which are then combined with a video file [9]. Examples of the types of log files created by typical applications used to analyse the consultation are shown in Figure 2.

Our current experimental work includes log file data from one or more of the following – with varying combinations used for different types of study. The specification dictated the need to be able to work with at the very least:

1. User Action Recorder (UAR) – An open source utility, and part of the ALFA toolkit, which collects all key strokes and mouse coordinates. It allows us to compare the number



Microsoft Excel - SDCforLFA_MCV_1...

	A	B	C	D
1	Starttime	Endtime	Event typ	Event value
2	14.6	79.6	ODC	30
3	27.4	93.1	ODC	29
4	60.3	90.2	ODC	12
5	108.3	128.5	ODC	4
6	128.5	152.6	ODC	29
7	152.6	204.4	ODC	4
8	204.4	232.2	ODC	29
9	427	920.9	ODC	3
10	893.2	11655.9	ODC	13
11	1465	1504.3	ODC	29
12	1551.3	1686.4	ODC	4
13	1944	2044.7	ODC	5
14	2262.5	2564	ODC	5
15	4245.3	4466.8	ODC	3
16	4741.9	5582	ODC	30
17	5808.9	5850.6	ODC	29
18	5857.4	5880.9	ODC	29
19	5888.4	5918.8	ODC	29
20	5964.8	5986.5	ODC	7

Keyfile - Notepad

```
15:10:53  :  (Enter)
15:10:54  :  (DownA)
15:10:55  :  (Enter)
15:10:56  :  (Mouse Click)
15:11:03  :  (Mouse Click)
15:11:07  :  (Mouse Click)
15:11:08  :  (Mouse Click)
15:11:09  :  (Mouse Click)
15:12:24  :  (Mouse Click)
15:12:25  :  (Mouse Click)
15:12:26  :  s
15:12:26  :  s
15:12:28  :  D
15:12:28  :  I
15:12:28  :  C
15:12:29  :  L
15:12:29  :  O
15:12:29  :  F
15:12:29  :  E
15:12:30  :  N
15:12:30  :  A
15:12:30  :  C
```

Microsoft Excel - sdl_jn05_09_logEvent_revised

	A	B	C	D	E	F
1	Time Stamp	Frame	Parameter	Parameter	Parameter	Parameter
2	Milliseconds	Rectangle	One	Two	Three	Four
3	1000	0	9	200	11	3
4	1000	2	101	333	420	5
5	1000	3	241	535	387	8
6	2001	3	5	67	8	1
7	4003	0	19	201	25	3
8	4003	2	3	67	12	1
9	4003	3	297	334	482	5
10	5003	0	303	800	412	12
11	5003	2	160	334	667	5
12	5003	3	1377	1000	2231	15
13	6004	0	58	467	76	7
14	6004	2	333	801	1395	12
15	6004	3	1037	1001	1679	15
16	7005	2	14	134	58	2

Figure 2. Log files created by ODC, User Action Recorder (UAR) for keyboard and mouse, and PRS.

of key strokes and length of mouse movement needed to complete common tasks on different electronic patient record (Electronic Patient Record (EPR) systems).

2. Voice Activation Recording – This is another open source utility from the ALFA toolkit, which time-stamps the start and end of speech. We use it where we want to precisely link transcript text to the video and other files. We have subsequently used this in 'Think aloud' protocol where computer users have described why they used the computer in the way they did, as well as making consultations available for conversational analysis.
3. Pattern Recognition Software (PRS) – this proprietary Algol^{TM} application measures pixel change and automatically. It has the allure of being able to automatically monitor keyboard activity; and positive body language. It will measure affirmative nodding towards the patient and the stillness or responsive behaviour of the patient [13].

4. Observational Data Capture (ODC) – There is nearly always a need to code behaviours or items of interest in a consultation manually. This is done using another component of the ALFA toolkit the ODC tool.
5. Other log files. It is hard to define how many log files a given study might require. We did this because there may be a need to nest other studies on top of existing analysis (e.g. our study of prescribing alerts) and we are unsure how many other utilities might emerge. For example, we have experimented with an electronic compass on the clinicians chair (most clinicians have typist type swivel chairs) and using a magnetic switch to monitor the chair angle as a surrogate for when the clinician is orientated towards the computer. A consensus emerged that it was better to go for a larger number; we opted for 12, rather than restrict analysis. However, the open source nature of the application means that we can add the ability incorporate larger number of files if required.

The LFA tool had to be able to link the multiple log files collected in analysis to a single medial file. The media component we use is a combined video output from multiple cameras; although this may appear complex it is relatively simple to do and is inexpensive [14]. The use of four channels was arrived at as part of our development process [4]. Our observational component uses four channels of video recording; three video cameras covering different views of the consultation and computer screen capture software. However, in the specification we stipulated that the number and type of media channels can be altered according to the requirements of any given study. A single video channel giving an overview of the consultation, or other activity; or just screen capture and audio, or just audio should be usable with the LFA application.

The major obstacle issue identified by researchers, which the LFA tool had to overcome, was the time taken in file management and processing time and inability to create a single analysable output when conducting consultation research. The existing manual methods take 14–35 h to analyse 1 h of consultation recording. The majority of this time is spent navigating the multi-channel video files to locate occurrences of particular activity or reviewing multiple log files to identify corresponding interactions. However, researchers also reported problems with managing large numbers of files relating to a single consultation. These included: lost files and incorrect labelling of files.

Finally, we wanted outputs which could be used for a number of different research purposes. The prime research project driving the need for this development was the wish to study the differences between the way that different brands of computer system interact in the clinical consultation. The aim of this work was to identify the best features of current systems in a way that system developers could use to develop better systems. We used another part of UML – the UML sequence diagram – to do this and we wanted to have an output from the LFA which could automatically generate as much of the UML sequence diagram as possible [15]. The consensus was that an extensible mark-up language (XML) was the best option.

3.2. Model developed using UML

UML class diagrams defined the design and implementation specifications for the tool. Each specification class diagrams were further divided into two main groups, as application and presentation classes. The application classes describe processing steps. They have two main processing entities, one to convert different log files into a common format and the second entity to combine these into a single file. The presentation classes illustrate the user interface aspect analogues to each application class.

Two UML class diagrams summarise the model developed and defined the design and implementation specifications for the LFA tool; Figure 3 shows the implementation classes. These diagrams also separated the application and presentation components. Application classes have two main processing functions: one to convert different log files into a common format and the other to combine these into a single file. These are further developed into subgroups; each specialised to handle the unique characteristics of different log files. These sub-groups are extensible should we wish to add additional log files at a future date.

3.3. Developing the data processing functionality and associated interface

Processing functions are classified into five sequential steps; (1) 'Settings', configuring the aggregation parameters, (2) 'Conversion', converting different log files into a common format, (3) 'Aggregation', combining common format log files together, (4) 'Display', viewing the aggregated output and (5) 'Output', exporting the output as a data file. The application's interface also has five panels displayed in sequence corresponding to each processing step, an example from step 3 is shown in Figure 4.

There are five re-usable Java classes representing each of the main processing steps. We reused the basic functionality of the conversion and output modules and extended their capabilities to handle log files or to produce the output in different formats. For example, the basic functionality of the conversion module initially developed to process the UAR logs (keyboard and mouse use) was re-used to process the ODC (manual coding) log files.

3.4. Displaying aggregated log file data using LFA

The final analysable output from the aggregated data could be obtained in either a tabular or one of two graphical formats. The tables give the sequence or frequency. The first graphical output option generates histograms of activity durations or frequencies. The graphical format represents all observed consultation activities as a on a single screen. Coded variables are columns with time running down the page along the y axis. Occurrences and durations appear as rectangles in each variable – we call each rectangle an 'Activity segment' and the whole table an 'Occurrence graph'. Within the occurrence graph variables can be re-ordered, and unwanted data deleted from the occurrence graph, though not from the underlying data. Examples of each are shown in Figure 5.

We also developed of a functional module to link the video player linked to the activity segment rectangles in the occurrence graph. This enables the analyst to click on any activity segment (or rectangle) on the occurrence graph and instantly go to that section of the media (Figure 6). In our subsequent use of this application the link is to the multi-channel video. This is particularly useful when comparing things which rarely occur in a consultation – like drug alerts.

3.5. Exporting the aggregated log files

The LFA has been designed to export the aggregated log file into XML (Extensible Mark-up Language) format. Many external analysis tools and statistical packages can interpret this XML version of the aggregated data, making the sharing of observational data much more efficient. An example of an XML output from the LFA in shown in Figure 7.

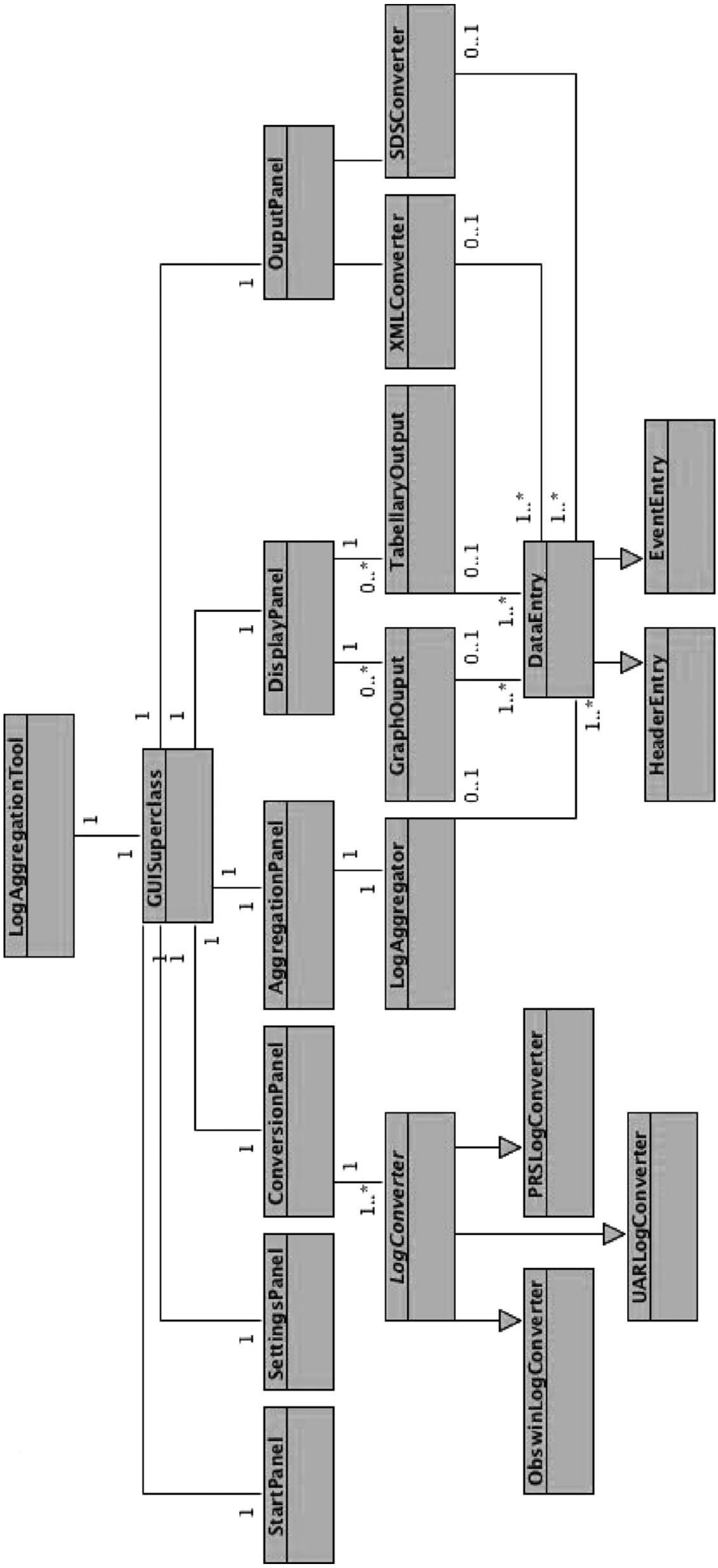


Figure 3. The UML implementation class diagram design for the LFA tool.

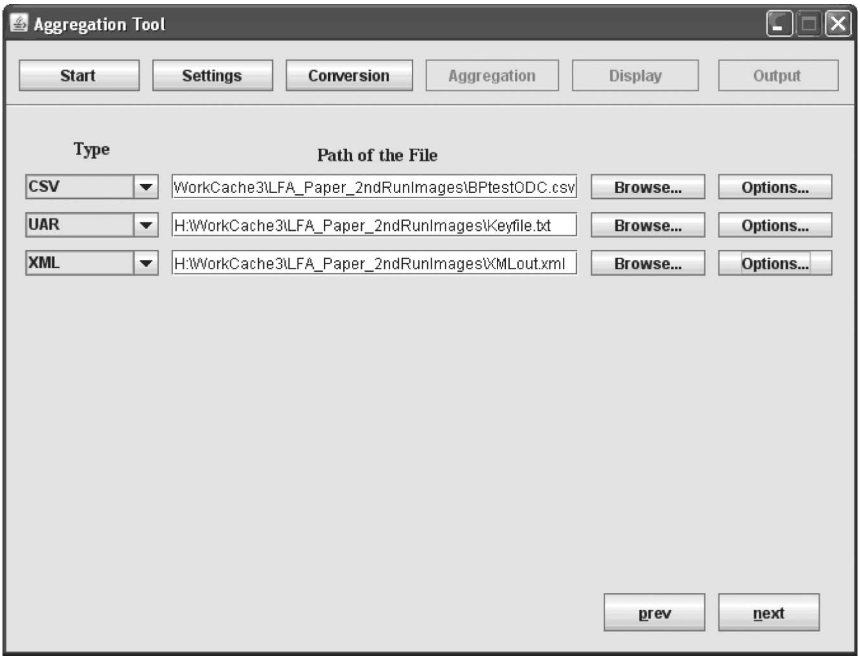


Figure 4. Example of the LFA interface: step 3 – aggregation.

3.6. Testing the LFA tool

Application testing sessions identified the lack of error detection routines. Most of them were due to the incomplete data recorded in the log files. New error messages were introduced to indicate these situations with detailed instructions. Testing stage also suggested additional features that would improve the usability of the software. Subsequently, features to zoom the graphical outputs, save them as image files, ability to add or remove variables in run time and ability to change the display colours were introduced.

4. Discussion

We have successfully developed the LFA tool – an application which aggregates log file output from disparate observational tools into a single format. Object-oriented design methods supported the development of modular software, and allow its extensibility in the future. This application removes the risk of human error in manual linkage of data, shortens analysis time and allows new types of analysis.

Although this application appears functional, more extensive testing may identify errors. There is scope to improve the application's interface and supporting documentation. This tool is now available as part of the ALFA open source toolkit available to download free for non-commercial use [16].

Video recording is an established technique for consultation analysis; a single camera setup is commonly used for assessment of clinicians in training [17]. Other researchers have used this technique to examine sociological aspects of computer-mediated consultations [18]; to measure the effect of computer system features or to study clinical practice [19,20]. Our view is that current methods do not provide sufficient details for robust analysis.

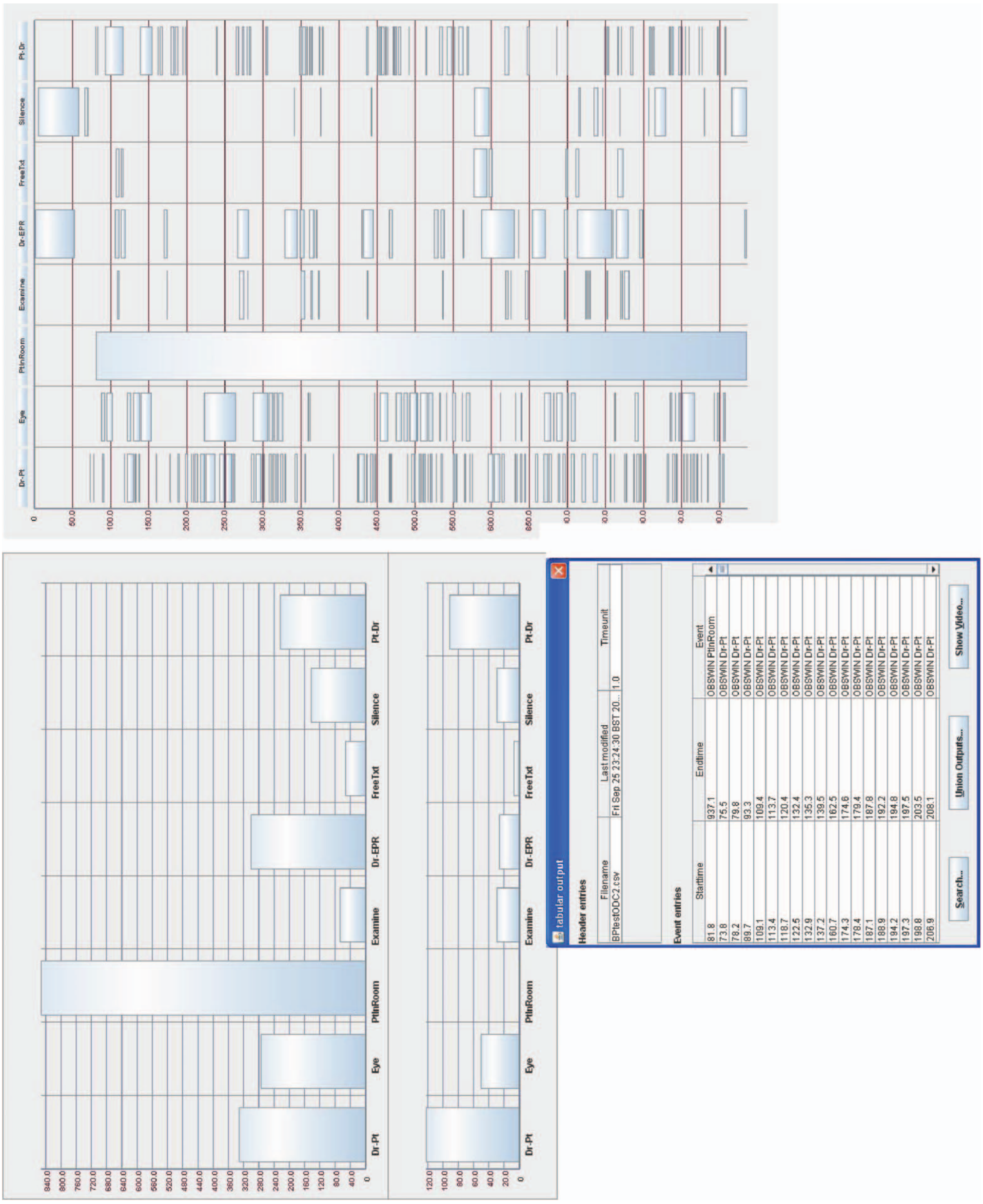


Figure 5. LFA tabular and graphical summary outputs from the LFA tool.

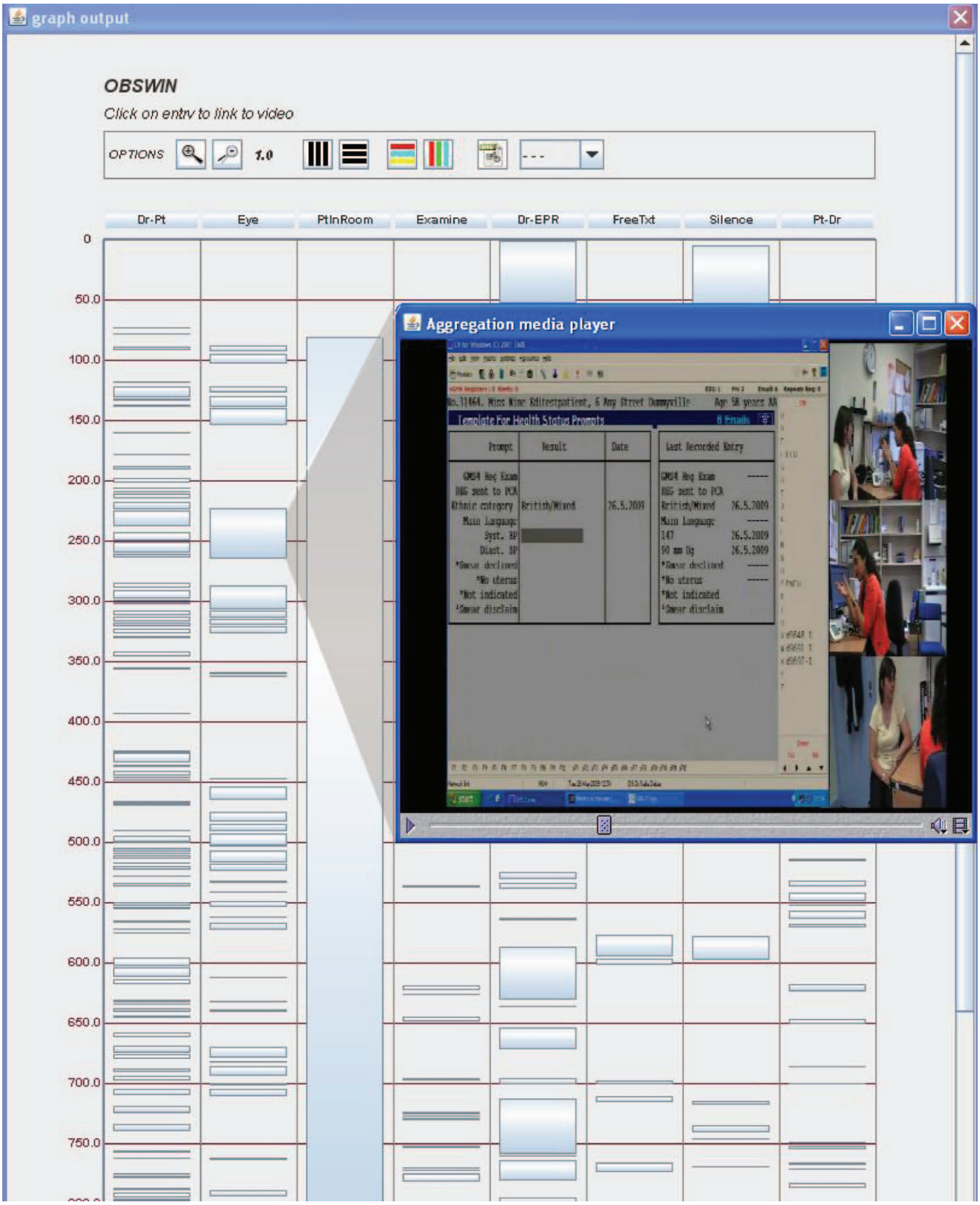


Figure 6. A mouse 'click' on an activity segment (any rectangle) on the occurrence graph links to the relevant place of the media file.

To date human–computer interaction (HCI) techniques have not been extensively used in healthcare settings; or as a routine part of system development. Established HCI analysis techniques use a combination of two or more data gathering techniques [21,22]. These include: gaze recognition, computer use, think aloud method and audio. Often they are

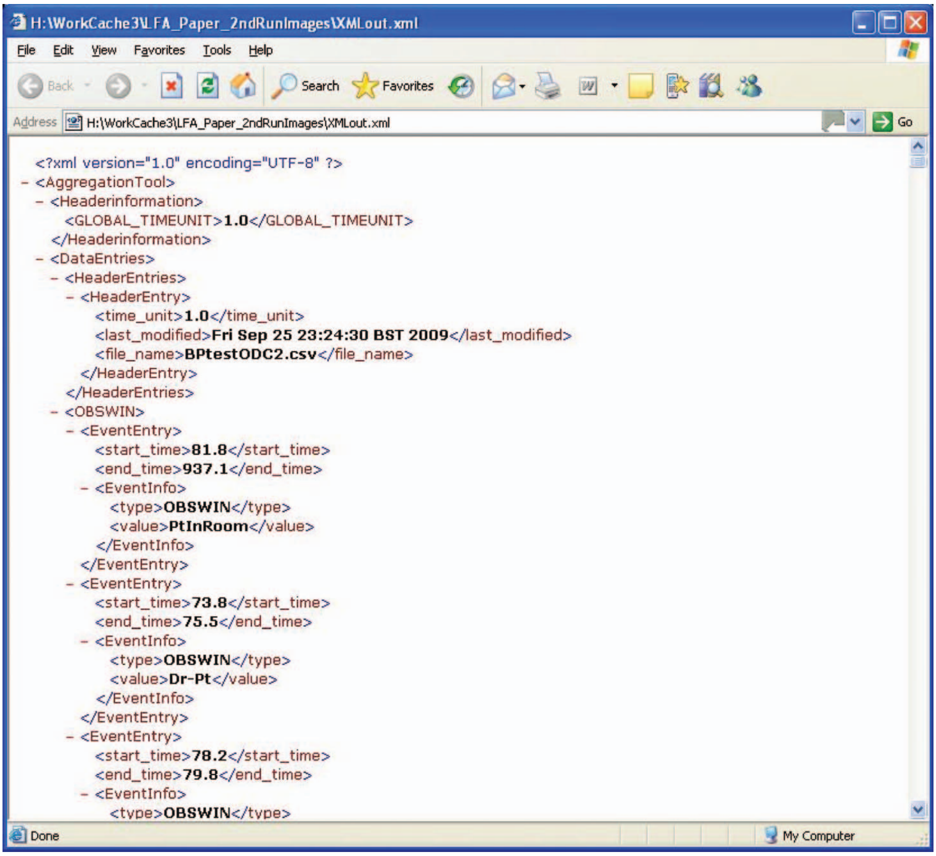


Figure 7. XML output from the aggregated log files.

integrated into a single video file with multiple visual data streams [23]. Most of the clinical information system assessment researches are based on qualitative assessment techniques, where video recording methods are utilised, they are assessed by one or more researchers reviewing the video [24]. Some of the HCI techniques may have less relevance in a scenario where the IT is supporting clinician–patient interactions. In clinical consultation we most often see the patient's attention and focus is primarily on the clinician – a dyadic relationship; with only transient triadic moments during which the clinician and patient share the computer screen [25].

This method has the potential to open a new paradigm in consultation research; the tool has performed in a robust way over its first 15 months of use. It is now possible to observe how the computer interacts with the actors in the consultation. The LFA needs to be used in the evaluation and research of how the computer influences the consultation. We hope the use of this tool leads to the development of systems which are more easily incorporated into the clinical workflow.

5. Conclusions

Separation of application logic from presentation logic has enabled the development of this consultation data processing tool. Its modular architecture supports the integration of future observation techniques, the export of results into universally comprehensible formats, and the extension of its functionality. This application combines different measures of doctor–computer–patient interactions enabling more effective analysis of the influence of the computer within the clinical consultation.

Acknowledgements

The authors thank the members of the Biomedical Informatics research team at St. George's University of London who participated in this software development project.

References

1. de Lusignan S, Chan T. The Development of Primary Care Information Technology in the United Kingdom. Journal of Ambulatory Care Management 2008;31:201.
2. Ash JS, Bates DW. Factors and forces affecting EHR system adoption report of a 2004 ACMI discussion. Journal of the American Medical Informatics Association 2005;12:8–12.
3. Schade CP, Sullivan FM, de Lusignan S and Madeley J. e-Prescribing, efficiency, quality: lessons from the computerization of UK family practice. Journal of the American Medical Informatics Association 2006;13:470–475.
4. Leong A, Koczan P, de Lusignan S, Sheeler I. A framework for comparing video methods used to assess the clinical consultation: a qualitative study. Medical Informatics and Internet in Medicine 2006;31:255–265.
5. de Lusignan S, Wgells SE, Russel C, Bevington WP, Arrowsmith P. Development of an assessment tool to measure the influence of clinical software on the delivery of high quality consultations. A study comparing two computerized medical record systems in a nurse run heart clinic in a general practice setting. Medical Informatics 2002;27:267–280.
6. Moulene MV, de Lusignan S, Freeman G, van Vlymen J, Sheeler I, Singleton A, Kumarapeli P. Assessing the impact of recording quality target data on the GP consultation using multi-channel video. Stud Health Technol Inform 2007,12(Part 2):1132–1136.
7. Refsum C, Kumarapeli P, de Lusignan S, Gunaratne A, Dodds R. Measuring the impact of different brands of computer system on the clinical consultation: a pilot study. Inform Prim Care 2008;16(2):119–127.
8. de Lusignan S, Wilson E, Dyble A, Grant T, Theadon A, Chan T. A strategy for using change recognition technologies to automate the assessment of patient-centered behavior in the clinical consultation; and the influence of computer use. Medinfo 2004;2004(CD):1569.
9. de Lusignan S, Kumarapeli P, Chan T, Pflug B, van Vlymen J, Jones B, Freeman GK. The ALFA (activity log files aggregation) toolkit: a method for precise observation of the consultation. Journal of Medical Internet Research 2008;10:e27.
10. Stallman R. GNU operating system: general public license (GNU GPL) [Internet]. Available from: URL: http://www.gnu.org (accessed 9 February 2009).
11. Object Management Group (OMG). Unified modelling language [Internet]. Available from: URL: http://www.uml.org (accessed 9 February 2009).
12. Kruchten P. The rational unified process: an introduction. 2nd ed., Reading, MA: Addison-Wesley, Inc.; 2000.
13. de Lusignan S, Wilson E, Dyble A, Grant T, Theadom A, Chan T. The feasibility of using pattern recognition software to measure the influence of computer use on the consultation. BMC Medical Informatics and Decision Making 2003;3:12.
14. Sheeler I, Koczan P, Wallage W, de Lusignan S. Low cost method three-channel video assessment of the consultation. Inform Prim Care 2007;15(1):25–31.
15. Kumarapeli P, de Lusignan S, Koczan P, Jones B, Sheeler I. The feasibility of using UML to compare the impact of different brands of computer system on the clinical consultation. Informatics in Primary Care 2007;15:245–253.
16. Kumarapeli P, de Lusignan S. The ALFA open source toolkit [Internet]. Available from: URL: http://www.biomedicalinformatics.info/alfa/ (accessed 9 February 2009).
17. McKinstry B, Walker J, Blaney D, et al. Do patients and expert doctors agree on the assessment of consultation skills? A comparison of two patient consultation assessment scales with the video component of the MRCGP. Family Practice 2004;21:75–80.
18. Pearce C, Dwan K, Arnold M, Phillips C. Analysing the doctor–patient–computer relationship: the use of video data. Informatics in Primary Care 2006;14:221–226.

19. Ram P, Grol R, Rethans JJ, Schouten B, van d, V, Kester A. Assessment of general practitioners by video observation of communicative and medical performance in daily practice: issues of validity, reliability and feasibility. Medical Education 1999;33:447–454.
20. Weinger MB, Gonzales DC, Slagle J, Syeed M. Video capture of clinical care to enhance patient safety. Quality & Safety in Health Care 2004;13:136–144.
21. Blandford A, Vanderdonckt J. People and Computers XV – Interaction Without Frontiers. Proceedings of HCI 2001. London: Springer; 2001.
22. Zhang Q, Imamiya A, Mao X, Go K. A gaze and speech multimodal interface. Proceedings of Distributed Computing Systems Workshops, Tokyo; 2004.
23. Oviatt S, Coulston R, Tomko S, et al. Toward a theory of organized multimodal integration patterns during human–computer interaction. Proceedings of the Fifth international conference on Multimodal interfaces, Sydney; 2003.
24. Kushniruk A, Patel V. Cognitive and usability engineering methods for the evaluation of clinical information systems. Journal of Biomedical Informatics 2004;37:56–76.
25. Pearce C, Dwan K, Arnold M, Phillips C, Trumble S. Doctor, patient and computer – a framework for the new consultation. Int J Med Inform 2009 Jan;78(1):32–38.