A Qualitative Study for the Design of an Integrated Development Gestural Environment for Task Flow Modelling Diagrams

José Angel Quintanar Morales, Carlos Alberto Fernández-y-Fernández and Mario Alberto Moreno Rocha

Universidad Tecnológica de la Mixteca, km. 2.5 carretera a Acatlima, Huajuapan de León, Oaxaca, México
{joseangel, caff, mmoreno}@mixteco.utm.mx

Abstract. The design of tools for software development still have improvements on the user experience side of it, for that reason it would be beneficial to experiment with a natural interaction paradigm. In order to do this, a first step is required to understand the environment and the real needs of software developers as well as qualitative research by improving the proposal obtained by a gestural Development Environment.

Keywords: IDEG, UsaLab, natural interaction, human factor, software development.

1 Introduction

Software development is a totally intellectual activity [16], it should pay special attention to the tools which support software developers, the most important, the integrated development environment (or Integrated Development Environment IDE). Despite all of the above, for this tool and other development tools, there is no significant progress in terms of adaptation and use of new technologies to diversify the user experience [1]. To raise the project proposal is imperative to first analyse the origins and trends IDEs.

According to the studies presented in [7], it is difficult to measure a developer's productivity. While it is true, there are methodologies such as Personal Software Process, which provides metrics to know time and productivity [5], although this neglects the human factor. Therefore, we should put more emphasis on other aspects, such as teamwork (beyond Team Software Process metrics) and improving the user experience. In the latter case, the tools on which rests the developer. For that reason our first step would be to know what do software developers think about it.

2 IDE's Background

The use of IDEs go back to the early 70's, it became popular when the need to develop complex software arose. Software prior to that time (late 60's) was designed using flowcharts and subsequently implemented by punched cards or paper tape, so there was a lack of support for a review by a compiler [12]. It wasn't until 1975 when the first IDE history, Master I [11] was developed by SoftLab in Munich. Master I consisted of a hybrid arrangement of hardware and software. As input means using a similar current computer keyboard; thus it was possible to input a fully intuitive manner.

IDE's projects, focusing on integrating complex and powerful tools available in modern IDE 's, in most cases, are not fully utilised. No one takes into account the capabilities of visual languages [13].

3 Visual Programming

The visual programming is commonly defined as the use of visual expressions (such as graphics, animation or icons) in the process of programming [4].

The visual programming objective is to improve the understanding of programs and simplify programming itself. The VPL (Visual Programming Language) can be classified according to the type and extent of visual expression used, such as languages icons, forms-based languages and diagrams language. The integrated visual programming environments provide graphics or icons that could be manipulated by users in an interactive way according to some specific spatial grammar for program construction [8].

4 User Interface

An important amount of devices having a form of natural interaction, have appeared implementing a natural user interface (NUI for its acronym in English). Natural user interface is one in which the user interact with a system or application without using control systems or input devices, instead, gestures are used as input gestures performed with the hands or body, the latter being a control stick [10].

The objective of this work is to experiment with gestural interaction into the software used by developers, specifically in the IDE, that would apply when using a language

The objective of this work is to experiment with gestural interaction into the software used by developers, specifically in the IDE, when using a visual language and flowcharts tasks will be designed. So far, we haven't found applications for development that would have the form of interaction we propose to use, so by building the prototype we could perform this experiment.

4.1 New Technologies

Kinect is a device that facilitate gestural interaction; with the release of the Libfreenect free driver project, Kinect was used outside video games console environment, thus enabling new forms of interaction aimed at supplementing the classic peripherals, as the pointer and keyboard [9, 14, 15].

4.2 Interaction Paradigm Change

Today we are witnessing the gradual change occurring in the paradigm of human-computer interfaces, everyday devices without keyboards are improved and controlled with elements such as voice, touch and movement. Maybe a crucial as when it happened in the 60's when an element appeared called XY Position Indicator for a Display System, which evolved up to the pointer we use today. This device came to create the basis on which the all-graphic applications that we are totally accustomed today [1] were built.

While you can find some resistance to change of form of interaction, everyday technology advances in the design and construction of innovative, novel devices. However, despite all, built applications to develop software continue without showing major changes in terms of interaction are concerned.

5 Proposed Solution

Using a high-fidelity prototype built using a User Centered Design methodology to explore the gestural interaction by designing workflow diagrams is proposed.

For this purpose there is a Kinect (which provides the facility to perform visual recognition) as a supplementary mean of the new pointer and keyboard interaction. It is through this device that arm movements made by users would be captured and from this captures an answer which will be then displayed in the IDE.

6 User Analysis

One of the most important points is to know the environment and the people with whom you will work to properly build the software tool. Information on how to build software models is required, so a series of Focus Groups were conducted. To find the right people, we performed an analysis by which the individual profiles were created, also known as *Personas* [3]. In this context, *Personas* refers to a person profile and not to a specific human being, to avoid confusion the term "user profile" [3, 6] is used.

6.1 Profiles Definition

In order to properly present and document the profiles definition, a template was used (presented in Figure 1) [3]. The purpose of the use of the template is that the information collected for the profile of the person is consistent and was organized in a clear and simple way.

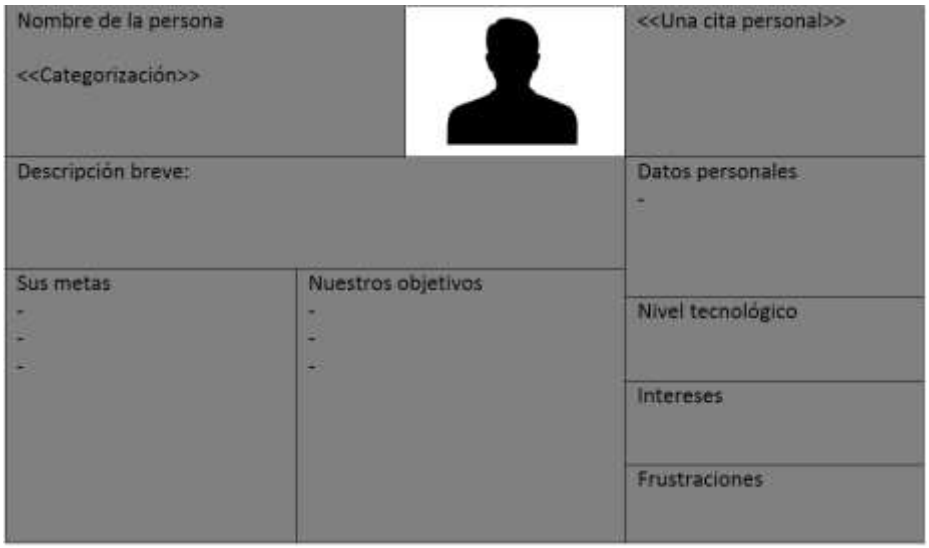

Fig. 1. Personas profile template.

Constructed profiles have been helpful in establishing priorities for the project, so far served to prioritise and / or add necessary features that users must follow to get the right information.

7 Focus Groups Development

The main objective of this study was to determine, define and experiment with gestural interaction through hand movements, while using a functional tool for the development of workflow diagrams. As a first step, we should know in depth the areas that only professional developers could provide. To identify those facets of interest, users contributed by given through knowledge and experience of people with different perspectives, but focused on the software development

Despite Focus Group has its origins in market research; it provides qualitative results by ability to interact with the target group being studied. In this way, it can be know and understand broadly the attitudes, needs, interests and motivations of the participants. In addition, Focus Groups follow the rigorous development of the research stages, starting with the problem statement and concluding with an oral and written presentation of research [6].

7.1 Description of Qualitative Research

For qualitative research three Focus Group sessions were conducted in the laboratory facilities of the *UsaLab Laboratorio de Usabilidad* at the *Universidad Tecnológica de la Mixteca* (UTM). Professional Developers who work in the two existing companies in the *Mixteca* region, attended the sessions, specifically in the city of Huajuapan de León, KadaSoftware SA de CV and VEUREKA SA de CV. Similarly master degrees students participated as well.

7.2 Instrument Design

Instruments required for the study are presented.

- Defining the problem: one must know the concerns and aspirations over the traditional way of creating software. To do this one start in the modelling phase, because that is where ideas involved in the projects are given. This is to know in depth how software modelling stage is done in actual practice and which chances are that they may decide to migrate to a tool that poses a new form of interaction.
- Study proposal: the opening subject will be how they begin a new project from an individual perspective. From this, the new questions and commentaries addressed the different phases of the professional life when was required to develop the system model, including the student stage and the small projects where the person was the only developer. After knowing the personal experiences and techniques we proceeded to begin the main subject of the focus group, which is that the developing team create a system model. The evaluated topics included the actual technique in which this activity is realized, the tools used and, the responsibilities of each team member. These two subjects are the basis of our research since provide the minimal necessary information.
- Session guide design and testing:

Below are the initial basic questions for the focus groups and the questions for the main subject.

1. How important is the modelling phase when developing a system individually?
2. How easy or difficult can be carried out that phase? Or can it be omitted?
3. Do you use some software tool to help with the modelling?
4. How important is the modelling phase when developing a project with a team?
5. How easy or difficult can be carried out that phase? Or can it be omitted?
6. How do you rate the development of a meeting of the development team for the modelling software?
7. In the meeting, the team leader is strict, bring forth participation, is purposeful?

7.3 Site Selection

The suitable site to carry out our focus group was the Usalab. It has specialized equipment such as video cameras, video edition, software, a Hessel camera (ideal for observation.) Additionally, it has monitoring equipment with IP cameras, digital recorders and equipment for measuring physiological signals. For these reasons and for its accessibility (within the university campus) it is the best option.

7.4 Selection of Participants

The selection process was realized considering employees from the software companies in the region. The basic criteria was as follows: study level, working directly as a developer, working in development teams, working as a team leader. It is noteworthy there was not economical or kind incentive for the participants in the focus group.

7.5 Focus Group

All were carried out at the premises of the Usalab of the UTM on a schedule of 19:30 to 20:30 hours. As instruments for each focus group, in addition to the session handbook, was presented a low-fidelity prototype as can be seen in Figure 2. This prototype was disclosed to the participants when explaining its characteristics and functionality.

The whole focus group process was recorded from two different angles. These video files were used to complement the final report.

7.6 Profile of the Participants

Following we describe the profile of each one of the participants of the focus group.

Miriam: computer engineer, postgraduate student of the master in applied computing program, 26 years old, junior programmer at KadaSoftware, 3 years of experience.

Carlos: computer engineer, postgraduate student of the master in applied computing program, 25 years old.

Noé: computer engineer, 27 years old, developer at Veureka S.A. de C.V., 3 years of experience.

Alfonso: computer engineering studies, 29 years old, developer at Veureka S.A. de C.V., 3 years of experience.

Erick: computer engineer, postgraduate studies of the master in applied computing program, 34 years old, technical leader at KadaSoftware, 8 years of experience.

Ana: computer engineer, 27 years old, analyst and developer at KadaSoftware, 4 years of experience.

Fig. 2. Low-fidelity prototype presented to the different focus groups [2].

8. Discussion of the Results

In this section we offer a detailed summary of the answers provides and the general attitudes of the people. The questions presented in section 7.2 are taken as a basis, these ones were exposed to the participants of the focus group.

8.1 How mportant is the modelling phase when developing a system individually?

The general answer was the expected "very important". Because it was possible to increase experience in software development and, at the same time, with the increasing of the complexity of the projects different strategies were adopted for the system modelling phase. It was evident that being personal projects there was no formal documentation related to the process. In most cases the modelling was simply an informal draft on paper to have a visual perspective and was not really defining for the development of the software.

8.2 How easy or difficult can be carried out that phase? Or can it be omitted?

In relation with this question none of the participants decided to skip the modelling system phase. Everyone carry out this activity with different level of formality.

8.3 Do you use some software tool to help with the modelling?

There was not a specific type of tool used to create the design when working individually. Basically the idea of a tool is having a visual description as a no formal guide to document the model.

8.4 How important is the modelling phase when developing a project with a team?

The participants indicated that this phase is imperative. Essentially because it is the part where they really get to know the details of the project. It was evident that this phase is carried out with a meeting, which could be informative or participative. In an informative meeting the team leader shows the architecture and the model that him or another team member propose and usually it is focused in technical details of the project. In a participative meeting the team leader presents the project and a set of sketches where the general solution of the software problem is depicted; the solution is discussed and improved by the team members. The team usually discuss technical aspects related to content and design, proposing at any time changes to the base model of the project.

8.5 How easy or difficult can be carried out that phase? Or can it be omitted?

There were a variety of opinions but in general the participants agreed that this phase has to be carried out in meetings as a team effort even if they shown some apathy.

8.6 How do you rate the development of a meeting of the development team for the modelling software?

For different reasons, the participants have some displeasure for the team meetings. The main identified reasons were the time loss, some disagreements with the requirements and a strict project approach.

Most of this displeasure was detected in relation with developers having a strict model for the meetings, whilst in the teams where the participation of all the members is encouraged, the developers believe their teams achieve a more precise model in less time.

8.7 In the meeting, the team leader is strict, bring forth participation, is purposeful?

In general the team leader encourages the participation of the team members. The exception is when the suggested modifications or improvements to the model are not under control to be accepted or rejected by the leader. In this case, the opinion of the development team is rejected. This kind of leader look for consensus on the technical aspect where the team can really take decisions.

9 Conclusions

From the opinions expressed by the participants of the focus group we could conclude that the development teams start out from a model when developing software, even when no specific process or methodology is used. Although sometimes no formal process is followed, the actions carried out in the meetings accomplish the general aim for this phase. Because the result from this modelling phase is useful as an approach to the solution, sometimes this solution is documented in order to have historical information of the project.

In addition, there seems to be a general consensus that the meetings are tedious and, for this reason, the proposal of improving the interactions and logic mechanisms in the work meetings were well received by the developers.

Unfortunately from the information provided, it was detected that the team members sometimes have no power of change the models presented by the team leader, in this cases they only can suggest implementation details.

As a result from the information analysis, we present in Figure 3 a prototype for the boardroom we will explore the improvement of the user experience, using the modelling software for task flow diagrams.

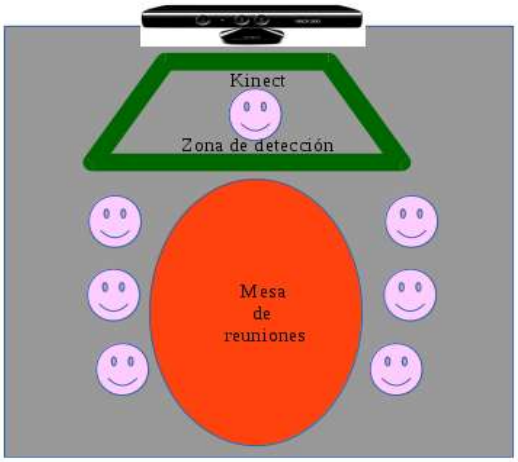

Fig. 3. Boardroom design for the low-fidelity prototype.

References

1. Cuevas , F. Interfaces de usuario alternativas. Software Guru 29, S2 (2010), 28–35.
2. Fernández-y-Fernández, C. A., and Quintanar-Morales, J. A. Integrated development environment gesture for modeling workflow diagrams. Tendencias en Investigación e Innovación en Ingeniería de Software: un enfoque práctico (2012).
3. Grosjea, J. C. A day in life of an agileux practitioner: Personas, Último acceso Enero 12 2014.

4. Hernández Valdemar , E. J., and Manuel, U. L. H. El paradigma de la programación visual. Fundación Arturo Rosenblueth (2013).
5. Humphrey, W. S. PSP: A Self-Improvement Process for Software Engineers. Addison-Wesley P., U.S.A, 2005.
6. Ivankovich-Guillén , C., and Araya Quesada, Y. Focus groups, técnica de investigación cualitativa en investigación de mercados. Ciencias económicas 29, 1 (2001), 545–554.
7. Johnson , L. F. On measuring programmer team productivity. Electrical and Computer Engineering, 1998. IEEE Canadian Conference on 2, 6127340 (1998), 701–705.
8. Johnson, W. M., Paul Hanna , J. R., and Millar , R. J. Advances in dataflow prog. lang. ACM Computing Surveys 36 (2004), 1–34.
9. Martín, H., Blake, J., and Machulis, K . Comunidad openkinect, Último acceso Julio 9 2013.
10. Mitra, S., and Acharya , T. Gesture recognition: A survey. Systems, Man, and Cybernetics, Part C: App. and Reviews 3, 37 (2007), 311–324.
11. Munden , I. B. M. G. Interaktives programmieren als systemsschlager, Último acceso Julio 3 2013.
12. Newman , P. S. Towards an integrated development enviroment. IBM Systems 21 (1982), 81–107.
13. Quintanar-Morales, J. A. Diseño e implementación de un plug-in para el modelado de diagramas de flujo de tareas. Universidad Tecnológica de la Mixteca, México, 2013.
14. Ramos, E., Ramírez, M., Nila, E., Figueroa , Diego , H. J., García , M., and Pérez , E. Based kinect application to promote mixtec culture. The 2013 Iberoamerican Conference on Electronics Engineering and Computer Science 7 (2013), 344–351.
15. Shotton, J. Real-time human pose recognition in parts from a single depth image. Microsoft Research (2010).
16. Walter, A.-R. El factor humano en los procesos del software. Software Guru 1, 33 (2011), 20–25.