Evaluating the Usability and the Communicability of Grid Computing Applications

Cristian Rusu, Silvana Roncagliolo, Arturo Figueroa
Pontificia Universidad Catolica de Valparaiso
Valparaiso, Chile
e-mail: cristian.rusu@ucv.cl, silvana@ucv.cl, arturo.figueroa.r@mail.pucv.cl

Virginica Rusu
Universidad de Playa Ancha
Valparaiso, Chile
e-mail: virginica.rusu@upla.cl

Dorian Gorgan
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
e-mail: dorian.gorgan@cs.utcluj.ro

***Abstract*—Usability is a main attribute of any interactive software system. Its relevance for Grid Computing applications is expected to increase, as the technical knowledge of grid users will gradually decrease. Usability evaluation for Grid Computing applications brings new challenges. A set of specific usability heuristics was defined and validated. The paper presents a Grid Computing communicability study and evaluates the communicability's impact on applications' usability.**

Keywords-usability; communicability; grid computing applications; semiotic engineering; usability heuristics.

I. INTRODUCTION

Grid computing is a relatively new, distributed computing technology, which relies on the coordinated use of different types of computing resources of an unspecified number of devices. The ISO/IEC 9241 standard defines the usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [1]. It is expected that the technical knowledge of grid users will gradually decrease, therefore the usability of Grid Computing applications will soon become a main issue.

Usability evaluation for applications based on emerging information technology brings new challenges. Is it the classical concept of usability still valid? Which are the dimensions of the (new) usability? How can it be measured? How should we develop for (better) usability? There is a need for new evaluation methods or at least for the use of traditional evaluations in novel ways [2].

The communicability is defined as the distinctive quality of interactive computer based systems that communicate efficiently and effectively to users their underlying design intent and interactive principles [3]. Communicability has (potentially) a major impact on system's usability.

A set of 12 usability heuristics for Grid Computing applications and an associated usability checklist were defined and validated in several case studies [4]. Later on, a semiotic inspection was performed, in one case study. The paper explores the communicability's impact on applications' usability. Section 2 summarizes the usability heuristics proposal for Grid Computing applications. Section 3 shows the results of the semiotic inspection and highlights the relationship between application's usability and communicability. Conclusions are presented in Section 4.

II. USABILITY IN GRID COMPUTING APPLICATIONS

Grid Computing users, their knowledge and specific tasks may be categorized as follows [5]: (1) service end-user, (2) service end-user execute, (3) power user agnostic of grid resource nodes, (4) power user requiring specific grid resource nodes, (5) power user developing a service, (6) service provider, (7) infrastructure system administrator. It is expected that the technical knowledge of grid users will gradually decrease. The number of users belonging to the first and the second of the above mentioned categories are growing fast. That is why we think the usability of Grid Computing applications will soon become a main issue.

Heuristic evaluation is a widely used inspection method [6]. A group of evaluators inspect the interface design based on usability principles (heuristics), usually Nielsen's ten heuristics [7]. Heuristic evaluation is easy to perform, cheap and able to find many usability problems (both major and minor problems). However, it may miss domain specific problems. That is why the use of appropriate heuristics is highly relevant.

In order to develop specific usability heuristics for Grid Computing applications, a 6 steps iterative methodology was followed [8]. A set of 12 usability heuristics and an associated usability checklist of 42 items were defined [4]. The 12 heuristics were grouped in three categories: (1) *Design and Aesthetics*, (2) *Navigation* and (3) *Errors and Help*.

The set of 12 new Grid Computing usability heuristics were checked against Nielsen's 10 heuristics, using *GreenView* and *GreenLand* as case studies [9], [10], [11].

 ISBN: 978-1-61208-177-9

Grid Computing heuristics worked better than Nielsen's heuristics, in both cases [4].

III. COMMUNICABILITY IN GRID COMPUTING APPLICATIONS

The *Special Interest Group on Computer-Human Interaction* (SIGCHI) of the *Association for Computing Machinery* (ACM) defines *Human-Computer Interaction* (HCI) as the discipline concerned with the design, evaluation and implementation of interactive computing systems for human use, and with the study of major phenomena surrounding them [12].

The *Semiotic Engineering* views HCI not just as a communication between users and software systems, but as a computer-mediated communication between designers and users, at interaction time. The system is therefore the designer's deputy, the artifact that transmits designer's intentions [13].

A. Semiotic Engineering and Communicability

The semiotic engineering considers HCI as an interactive and progressive communication process about how to communicate with the system, when, why, and to what effects. The software system speaks for its designers in various types of conversations, specified at design time. The process is one of communication about communication, or metacommunication. Communicability is the attribute that defines the quality of the metacommunication.

The content of the designer message can be paraphrased by a generic metacommunication template: "*Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision*" [14].

There are three distinctive classes of signs in the designer's deputy's interactive discourse:

- *Static signs*, whose meaning is interpreted independently of temporal and causal relations;
- *Dynamic signs*, which are bound to temporal and causal aspects of the interface, emerging with the interaction;
- *Metalinguistic signs*, which explicitly communicate to users the meanings encoded in the system and how they can be used.

Static signs stimulate the user to engage in interaction with the system; they help the user anticipate what the interaction will be like and what consequences it should bring about. Dynamic signs confirm or disconfirm the user's anticipation. The meaning of static and dynamic signs is explicitly informed by metalinguistic signs.

B. Communicability Evaluation

The semiotic engineering offers two methods to evaluate the quality of metacommunication in HCI:

- The *Semiotic Inspection Method* (SIM), and
- The *Communicability Evaluation Method* (CEM).

SIM explores the emission of metacommunication, seeking to reconstruct its content, expressions, and targeted receivers. CEM explores the reception of metacommunication, seeking to identify, by means of user observation, empirical evidence of the effects of the designers' messages as they are encountered at interaction time [14]. SIM is an inspection method, involving only specialized evaluators. CEM is a test method, involving real and/or representative users.

SIM aims to reconstruct designer's message using the metacommunication template as a guide. It includes five core steps:

1. The analysis of metalinguistic signs,
2. The analysis of static signs,
3. The analysis of dynamic signs,
4. A comparison of the designer's metacommunication message generated in the previous steps,
5. A final evaluation of the inspected system's communicability.

In steps 1, 2, and 3, the evaluator does a segmented analysis of the system, deconstructing the metacommunication message. It allows to inspection of how the designer communicates with each type of sign (each main communication channel). In steps 4 and 5, the evaluator reconstructs the metacommunication message (filling out the metacommunication template) by comparing, integrating, and interpreting the data collected in previous steps of the method.

C. The Semiotic Inspection of a Grid Computing Application

A semiotic inspection of *GreenView* was done. First, the metalinguistic signs were analyzed. They are limited to pop-up messages explaining functionality (Fig. 1), and some general explanation (Fig. 2).

The metacommunication template based on metalinguistic signs may be synthesized as follows: "*You are an expert user that knows what to do. You need to process large amount of specific data, efficiently, in order to get specific information. I have designed a specialized system, that you now how to use, therefore you don't need explanations*".

Static signs are abundant. They include specific layout, controls grouping, icons, and explicit menu options. Fig. 3 shows the "*Fine-to-Coarse*" control panel, highlighting controls grouping (green color), icons and menu options (red color).

The metacommunication template based on static signs may be synthesized as follows: "*You are a user that knows what to do. You need to get easy access to specific functionality, to do actions efficiently, in order to perform specific processes. I have designed a specialized system, which you should be able to use efficiently*".

Dynamic signs include explicit information on system's status (Fig. 4), alternative options of (intuitive) direct manipulation (Fig. 5), and pop-up windows (Fig. 6).

The metacommunication template based on dynamic signs may be synthesized as follows: "*You perfectly know what the system offers. You need to work efficiently, choosing your own way. I have designed a specialized system, which offers alternative ways and feedback that you should understand*".

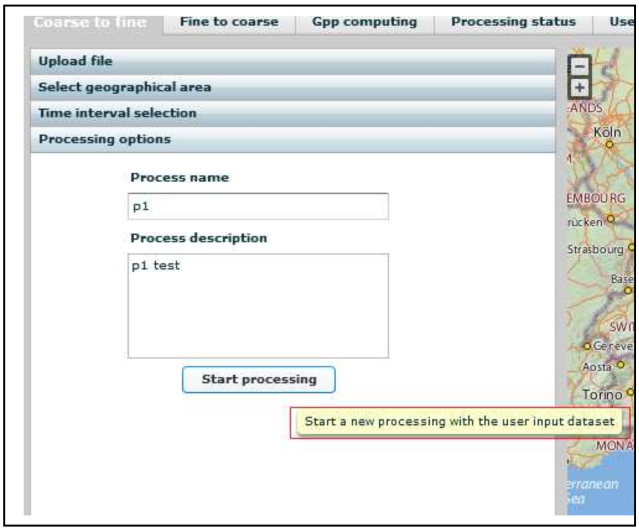

Figure 1. Pop-up messages explaining functionality.

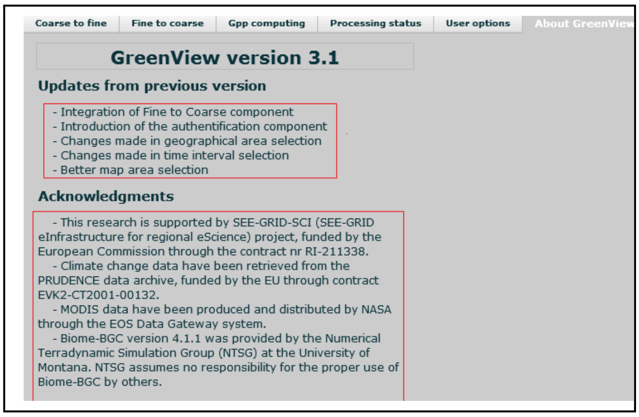

Figure 2. Explanations about *GreenView* application

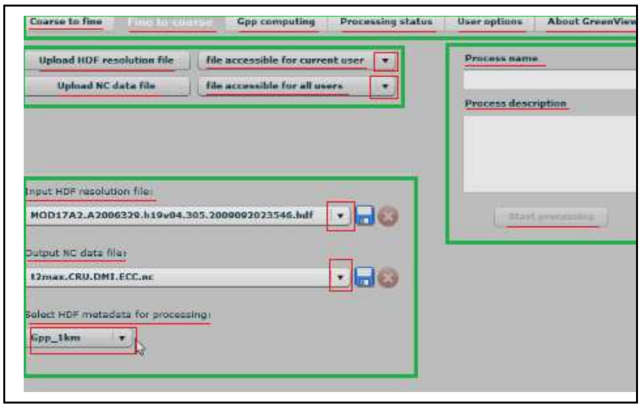

Figure 3. Static signs on "*Fine-to-Coarse*" control panel.

The three metacommunication messages generated by metalinguistic, static and dynamic signs are homogeneous. Communication strategies based on static and dynamic signs are quite similar and coherent. The metacommunication message generated by metalinguistic signs is rather cryptic. Metalinguistic signs are scarce. Textual static signs are themselves (a kind of) metalinguistic signs.

Figure 4. Explicit feedback on processing status.

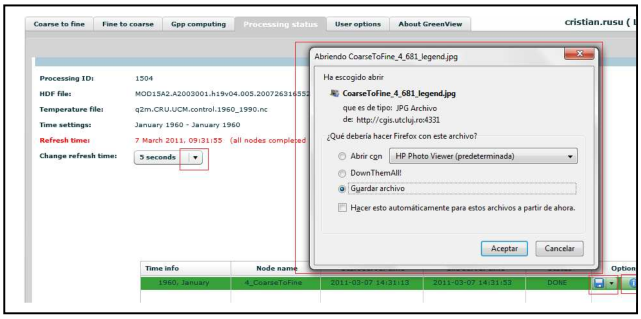

Figure 5. Alternative direct manipulation options.

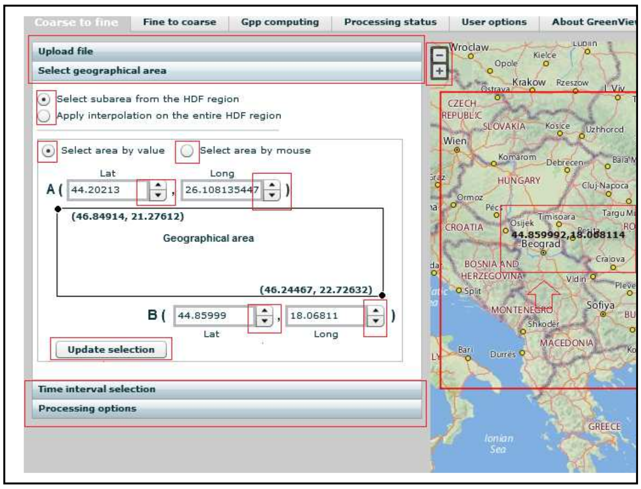

Figure 6. Pop-up windows.

The overall evaluation of *GreenView*'s communicability shows that the system is oriented to expert users. The system is still under development. New functionality is added, in new system's versions. Efficiency and flexibility are explicit system's goals.

D. Usability and Communicability in GreenView

Table 1 shows the number of usability problems identified in *GreenView*. A significant amount of problems were associated to *Errors and Help* heuristics: H10 (*Error prevention*), H11 (*Recovering from errors*), and H12 (*Help and documentation*). As the semiotic inspections proved, there is a lack of metalinguistic signs in *GreenView*. Therefore such usability problems were somehow expected.

The metacommunication global message of *GreenView* highlights its focus on expert users. That could explain a relatively large number of usability problems associated to heuristic H1 (*Clarity*). As the metacommunication global message shows, *GreenView* also focus on efficiency and flexibility. These goals seem to be accomplished, as a low number of usability problems were associated to heuristics H5 (*Consistency*), H6 (*Shortcuts*), H8 (*Explorability*), H9 (*Control over actions*), and no usability problems were associated to heuristic H7 (*Low memory load*).

An evident relationship between application's usability and communicability may be observed. The global metacommunication message highlights application's goals, users' profile, and communication strategies, anticipating associated usability problems.

TABLE I. USABILITY PROBLEMS IDENTIFIED IN *GREENVIEW*

Usability Heuristic	Associated Usability Problems
H1: Clarity	3
H2: Metaphors	1
H3: Simplicity	0
H4: Feedback	2
H5: Consistency	1
H6: Shortcuts	1
H7: Low memory load	0
H8: Explorability	1
H9: Control over actions	1
H10: Error prevention	3
H11: Recovering from errors	2
H12: Help and documentation	2
Total:	**17**

IV. CONCLUSIONS

The current use of Grid Computing is at the hand of experts and researchers, but it is expected that in the future the technical knowledge of grid users will decrease. Research usually focuses on Grid Computing based application development from a technical point of view. There is a need for new usability evaluation methods or at least evaluations should be particularized for Grid Computing environments. A set of 12 Grid Computing usability heuristics and an associated usability checklist were specified, validated and refined by an iterative process.

The semiotic engineering brings a new perspective on human–computer interaction. Semiotic inspections identify the designer–to–user message reconstructing the metacommunication template. Communicability evaluation may anticipate associated usability problems. Therefore it may be a powerful tool, for researchers, usability professionals and Grid Computing application developers.

ACKNOWLEDGMENT

The authors would like to thank to all the participants involved into the experiments that the present study required. The work was highly supported by the *School of Informatics Engineering* of the *Pontifical Catholic University of Valparaiso* – Chile and by the *Computer Science Department* of the *Technical University of Cluj-Napoca* - Romania.

REFERENCES

[1] International Organisation for Standardisation, ISO 9241, "Software Ergonomics Requirements for office work with visual display terminal (VDT)", Geneva, 1998.

[2] C. Wiberg, K. Jegers, and H. Desurvire, "How Applicable is Your Evaluation Methods – Really?", Proc. 2nd International Conferences on Advances in Computer-Human Interactions (ACHI 2009), IEEE Press, ISBN: 978-1-4244-3351-3, 2009, pp. 324-328.

[3] C. de Souza C., "The Semiotic Engineering of Human-Computer Interaction", MIT Press, 2005.

[4] S. Roncagliolo, V. Rusu, C. Rusu, G. Tapia, D. Hayvar, and D. Gorgan, "Grid Computing Usability Heuristics in Practice", Proc. 8th International Conference on Information Technology: New Generations (ITNG 2011), IEEE Press, 2011, pp. 145-150.

[5] M. Norman, "Types of grid users and the Customer-Service Provider relationship: a future picture of grid use", University of Oxford, 2006.

[6] J. Nielsen, "Usability Engineering", Morgan Kaufmann Publishers, 1993.

[7] J. Nielsen, "Ten Usability Heuristics", www.useit.com/papers/heuristic/heuristic_list.html (retrieved: 11/11/2011).

[8] C. Rusu, S. Roncagliolo, V. Rusu, and C. Collazos, "A Methodology to Establish Usability Heuristics", Proc. 4th International Conferences on Advances in Computer-Human Interactions (ACHI 2011), IARIA, ISBN: 978-1-61208-003-1, 2011, pp. 59-62.

[9] R. Meszaros, I. Lagzi, Z. Barcza, and G. Gelybo, "GreenView Application Specifications", Eotvos Lorand University, Hungary, 2008.

[10] D. Gorgan, T. Stefanut, V. Bacu, and D. Mihon, "Grid based Environment Application Development Methodology", LNCS 5910, Springer, 2010, pp. 499-506.

[11] C. Rusu, S. Roncagliolo, T. Gonzalo, H. Danae, V. Rusu, and D. Gorgan, "Evaluating the Usability of Intercultural Collaboration Platforms: Grid Computing Applications", Proc. 3rd ACM International Conference on Intercultural Collaboration (ICIC 2010), ACM Digital Library, ACM 978-1-4503-0108-4/10/08, 2010, pp. 179-182.

[12] ACM's Special Interest Group on Computer-Human Interaction, www.sigchi.org (retrieved: 11/11/2011).

[13] H. Sharp, Y. Rogers, and J. Preece, "Interaction Design: Beyond Human-Computer Interaction", Wiley, 2007.

[14] C. de Souza and C. Leitão, "Semiotic Engineering Methods for Scientific Research in HCI", Morgan & Claypool, 2009.