Usability Heuristics for Grid Computing Applications

Cristian Rusu[1], Silvana Roncagliolo[1], Gonzalo Tapia[1], Danae Hayvar[1], Virginica Rusu[2], Dorian Gorgan[3]

[1]Pontificia Universidad Catolica de Valparaiso, Valparaiso, Chile
e-mail: cristian.rusu@ucv.cl, silvana@ucv.cl, g.tapia.aldea@gmail.com, dhayvar@gmail.com

[2]Universidad de Playa Ancha, Valparaiso, Chile
e-mail: virginica.rusu@upla.cl

[3]Technical University of Cluj-Napoca, Cluj-Napoca, Romania
e-mail: dorian.gorgan@cs.utcluj.ro

Abstract **— Usability evaluation for applications based on emerging information technology brings new challenges. Grid Computing is a relatively new, distributed computing technology, based on sharing different types of computational resources, located in various geographic locations. Technical knowledge of grid users is expected to decrease in the future; that is why the usability of Grid Computing applications will become a main issue. There is a need for new usability evaluation methods or at least for the use of traditional evaluations in novel ways. A set of heuristics is proposed and validated, in order to help the heuristic evaluations of Grid Computing applications.**

Keywords – usability; usability heuristics; grid computing applications

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, which are not necessarily at the same geographical location. The process is transparent for users, allowing the use of resources as a single supercomputer.

There are many projects worldwide making use of grid infrastructure, most of them for scientific purposes. Current research usually focuses on Grid Computing based application development from a technical point of view, rather than a user–centered approach. There is a necessity to establish methodologies that could lead to applications with a high level of usability. Such methodologies have to include accurate usability evaluations.

The usability evaluation of a software system is one of the most important stages in the user centered design approach. It allows obtaining the usability characteristics of a software system and the extent to which the usability attributes, usability paradigms and usability principles are being implemented [1].

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 paper focuses on usability evaluation of Grid Computing applications by heuristic evaluations. A set of 12 specific usability heuristics is proposed and validated. Section 2 highlights the basic features of Grid Computing applications and the challenge of their usability evaluation. Section 3 presents a proposal of usability heuristics, which validation is described in section 4. Conclusions are presented in section 5.

II. USABILITY IN GRID COMPUTING APPLICATIONS

Grid Computing applications aim to solve problems that usually require a large number of processing cycles, storage and access of large amounts of data, sometimes distantly located or administered by various organizations, access to specialized equipment, and inter-organizational collaboration of users.

Grid Computing is defined by a set of basic features: abstraction, resource sharing, flexibility, decentralized management and control, scalability, high performance, security, generalization, personalization, heterogeneity. The use of Grid Computing technology has significant advantages: allows independent administrative domains, offers a good cost/performance ratio, enables the sharing of multiple types of resources, allows the integration of heterogeneous systems and resources, offers great fault adaptability and the capacity of easily adding new resources or replacing old ones, to provide new features [3] [4].

Grid Computing cover a wide range of application fields, and it is particularly useful in science, where experiments, simulations, or other research need a power that cannot be offered by standalone supercomputers or clusters of isolated organizations. Some Grid Computing applications are processor-intensive; others may require massive storage.

Depending on the type of resources that are mainly used, the main types of grid are:

- *Computing Grid*: designed to provide as much computing power as possible.

- *Data Grid*: allows the management and sharing of huge amounts of distributed data.
- *Service Grid*: provides services that cannot be provided by a single computer.
- *Equipment Grid*: provides access to special type of equipment, not easily available, either because of their high cost, geographic localization, or other difficulties.

The use of Grid Computing applications includes:

- *Job Submission*: Users specify the definition of tasks to execute and data to use.
- *Monitoring*: A monitoring interface allows users to check the status of the processing.
- *Visualization*: When jobs processing has finished, a visualization interface shows the results.
- *Web Portals*: Nowadays many Grid Computing based projects offer access to their services through Web applications, by Web portals.

Grid Computing users, their knowledge and specific tasks may be categorized as follows [5]:

- *Service end-user*: low technical knowledge; data input and grid services user.
- *Service end-user execute*: some technical knowledge; job submission.
- *Power user agnostic of grid resource nodes*: high technical knowledge; application development.
- *Power user requiring specific grid resource nodes*: high technical knowledge; application development, aware of specific grid resources nodes.
- *Power user developing a service*: high technical knowledge; services development.
- *Service provider*: high technical knowledge; identity and authorization management.
- *Infrastructure system administrator*: high security and infrastructure knowledge; grid nodes system administration.

The current use of Grid Computing is at the hand of experts and researchers with extensive (specific and technical) knowledge. Most of the Grid Computing applications users have nowadays similar knowledge and similar background, but we may infer that such similarities will be no longer the rule.

It is expected that in the future the technical knowledge of grid users will decrease. The number of users belonging to the first and the second of the above mentioned categories is growing fast. That is why we think the usability of Grid Computing applications will become a main issue.

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 [6].

Usability is not a one-dimensional property of the interface; it is a combination of factors. Effectiveness refers to the accuracy level that the user achieves goals. Efficiency refers to the resources employed by the user to accomplish these goals. Finally, the satisfaction is related to the comfort of the user during the interaction with the software system.

Usability evaluation methods are commonly divided into inspection and testing methods. Inspection methods find usability problems based on the expertise of usability professionals. Testing methods find usability problems through the observation of the users while they use (and comment on) a system interface [7].

Heuristic evaluation is a widely used inspection method. A group of evaluators inspect the interface design based on the usability principles (heuristics). 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 significant.

Grid Computing has evolved from scripts to portals and Web interfaces, therefore usability heuristics for Grid Computing should be developed from this new perspective [8] [9] [10].

III. DEFINING GRID COMPUTING USABILITY HEURISTICS

In order to develop specific usability heuristics for Grid Computing applications the following steps were followed [11]:

- An *exploratory* stage, to collect bibliography related with the main topics of the research: Grid Computing applications, usability evaluation, and usability heuristics.
- A *descriptive* stage, to highlight the most important characteristics of the previously collected information, in order to formalize the main concepts associated with the research.
- A *correlational* stage, to identify the characteristics that the usability heuristics for Grid Computing applications should have, based on traditional heuristics and case studies analysis.
- An *explicative* stage, to formally specify the set of the proposed heuristics, using a standard template.
- A *validation* (experimental) stage, to check the new heuristic against traditional heuristics by experiments, through heuristic evaluations performed on selected case studies, complemented by user tests.
- A *refinement* stage, based on the feedback from the validation stage.

Based on the well known and widely used Nielsen's 10 heuristics and extensively analyzing several Grid Computing applications, especially *GreenView* [12], a set of 12 new usability heuristics was developed for heuristic evaluations of Grid Computing applications.

Grid Computing heuristics were specified using the following template:

- *ID, Name and Definition*: Heuristic's identifier, name and definition.
- *Explanation*: Heuristic's detailed explanation, including references to usability principles, typical

usability problems, and related usability heuristics proposed by other authors.

- *Examples*: Examples of heuristic's violation and compliance.
- *Benefits*: Expected usability benefits, when the heuristic is accomplished.
- *Problems*: Anticipated problems of heuristic misunderstanding, when performing heuristic evaluations.

The 12 proposed heuristics were grouped in three categories: (1) *Design and Aesthetics*, (2) *Navigation* and (3) *Errors and Help*. A summary of the proposed heuristics is presented below, including heuristics' ID, name and definition.

Design and Aesthetics Heuristics:

(H1) *Clarity*: A Grid Computing application interface should be easy to understand, using clear graphic elements, text and language.

(H2) *Metaphors*: A Grid Computing application should use appropriate metaphors, making the possible actions easy to understand, through images and familiar objects.

(H3) *Simplicity*: A Grid Computing application should provide the necessary information in order to complete a task in a concise (yet clear) manner.

(H4) *Feedback*: A Grid Computing application should keep users informed on the jobs' progress, indicating both the global and the detailed state of the system. The application should deliver appropriate feedback on users' actions.

(H5) *Consistency*: A Grid Computing application should be consistent in using language and concepts. The forms of data entry and visualization of results should be consistent.

Navigation Heuristics:

(H6) *Shortcuts*: A Grid Computing application should provide shortcuts, abbreviations, accessibility keys or command lines for expert users.

(H7) *Low memory load*: A Grid Computing application should maintain the main commands always available. It should offer easy to find elements, functions and options.

(H8) *Explorability*: A Grid Computing application should minimize navigation and should provide easy, clear, and natural ways to perform tasks.

(H9) *Control over actions*: A Grid Computing application should offer ways to cancel a running task or process. It should allow undo and/or changes of actions.

Errors and Help Heuristics:

(H10) *Error prevention*: A Grid Computing application should prevent users from performing actions that could lead to errors, and should avoid confusions that could lead to mistakes.

(H11) *Recovering from errors*: A Grid Computing application should provide clear messages, hopefully indicating causes and solutions of errors.

(H12) *Help and documentation*: A Grid Computing application should provide an easy to find, easy to understand, and complete online documentation. It should provide contextual help and glossary of terms for novice users.

Table 1 presents the mapping between Grid Computing 12 heuristics and Nielsen's 10 heuristics [13].

TABLE I. MAPPING BETWEEN GRID COMPUTING HEURISTICS AND NIELSEN'S HEURISTICS

Grid Computing Heuristics		**Nielsen's Heuristics**	
ID	***Definition***	***ID***	***Definition***
H1	Clarity	N2	Match between system and the real world
H2	Metaphors		
H3	Simplicity	N8	Aesthetic and minimalist design
H4	Feedback	N1	Visibility of system status
H5	Consistency	N4	Consistency and standards
H6	Shortcuts	N7	Flexibility and efficiency of use
H7	Low memory load	N6	Recognition rather than recall
H8	Explorability	N3	User control and freedom
H9	Control over actions		
H10	Error prevention	N5	Error prevention
H11	Recovering from errors	N9	Help users recognize, diagnose, and recover from errors
H12	Help and documentation	N10	Help and documentation

Heuristics H1 and H2 particularize Nielsen's N2 heuristic. Heuristics H8 and H9 denote Nielsen's N3 heuristic. Both N2 and N3 heuristics where detailed and particularized based on the characteristics of the Grid Computing applications, their evolution from scripts to Web interfaces, and the new (heterogeneous) type of users they have.

Heuristic H3 particularizes Nielsen's N8 heuristic, emphasizing the complex tasks that Grid Computing users have to deal with. Heuristic H4 particularizes Nielsen's N1 heuristic into the context of Grid Computing applications, detailing specific feedback requirements. As there are not yet widely recognized standards for Grid Computing applications, heuristic H5 particularize Nielsen's N4 heuristic, stressing the dominance of the consistency over standards. Heuristic H6 provides more specific means than Nielsen's N7 heuristic, and heuristic H7 specifies more precisely Nielsen's N6 heuristic, based on the characteristics of the Grid Computing applications.

Finally, heuristics H10, H11 and H12 put Nielsen's heuristics N5, N9 and N10 (respectively) into the context of Grid Computing applications.

IV. VALIDATING GRID COMPUTING USABILITY HEURISTICS

The 12 proposed Grid Computing usability heuristic were checked against Nielsen's 10 heuristics, using *GreenView* and *GreenLand* as case studies. The potential of the Grid Computing heuristics was also checked in usability evaluations of Grid Computing applications as intercultural collaboration platforms.

A. Case Study: GreenView

GreenView is an environmental application that uses high-resolution satellite measurements in climate related studies, modeling the pollution and the impact that urban spaces have on vegetation (Fig. 1). As *GreenView* authors acknowledge, the development of environmental applications based on Grid infrastructures and dedicated to non-technical experts is a challenging task [14].

GreenView v3.1 was examined by two groups of 4 evaluators each. All 8 evaluators had similar (medium) experience in heuristic evaluations (with Nielsen's heuristics), but no experience in usability evaluation of Grid Computing applications. They all had comparable (low, if some) experience in using Grid Computing applications.

The first group performed a heuristic evaluation of *GreenView*, using only the 12 new (Grid Computing) heuristics (based on the full heuristics' specification). The second group performed a similar heuristic evaluation, but using only the Nielsen's 10 heuristics. Table 2 shows the number of usability problems identify by each group of evaluators.

TABLE II. NUMBER OF USABILITY PROBLEMS IDENTIFIED IN *GREENVIEW*, BY HEURISTICS

Group 1: Using Grid Computing Heuristics		**Group 2: Using Nielsen's Heuristics**	
ID	***Number of problems***	***ID***	***Number of problems***
H1	3	N2	1
H2	1		
H3	0	N8	3
H4	2	N1	2
H5	1	N4	1
H6	1	N7	0
H7	0	N6	0
H8	1	N3	1
H9	1		
H10	3	N5	2
H11	2	N9	1
H12	2	N10	1
Total:	**17**	**Total:**	**12**

When using Grid Computing heuristics, more usability problems were captured than using Nielsen's heuristics. Analyzing the total of 29 problems identified by the 8 evaluators, we found that:

- 11 problems (38%) were identified by both groups of evaluators,
- 12 problems (41%) were identified only by the group that used Grid Computing heuristics,
- 6 problems (21%) were identified only by the group that used Nielsen's heuristics.

The results seem to prove that Grid Computing heuristics work better than Nielsen's heuristics. However, the question that arises is: why 6 usability problems were not identified using Grid Computing heuristics? There are two possible reasons:

(1) Grid Computing heuristics were not able to identify these problems, either because there are no appropriate heuristics, or because the heuristics are not properly specified.

(2) Evaluators using Grid Computing heuristics subjectively ignored the problems.

The problems identified only by Nielsen's heuristics were associated to heuristics N8 - *Aesthetic and minimalist design* (3 problems), N1 – *Visibility of system status* (1 problem), N5 – *Error prevention* (1 problem), and N10 – *Help and documentation* (1 problem). The set of Grid Computing heuristics provides the tools that can potentially identify all these problems: H3 - *Simplicity*, H4 - *Feedback*, H10 - *Error prevention*, and H12 - *Help and documentation*, respectively. So, the first hypothesis is unlikely to be the true.

All 6 problems identified only by Nielsen's heuristics were qualified with relatively low severity scores (an average of 2.5 or less, on a five point scale). The second hypothesis seems to be the correct one.

In order to validate the second hypothesis, a usability test was designed and performed, with 5 users. The test was focused on the 6 usability problems identified only by Nielsen's heuristics. All these problems were not in fact perceived as real problems by users, so the second hypothesis was validated.

Analyzing the 12 problems identified only by Grid Computing heuristics, most of them were qualified as sever: 8 of 12 problems had an average severity of 2.5 or superior, on a five point scale. Moreover, 3 of 12 problems had (very high) average severity (3.25).

B. Case Study: GreenLand

GreenLand is an environmental application that processes high-resolution *Landsat* satellite images in order to obtain thematic maps of specific elements, such as land, water, air, and vegetation (Fig. 2).

GreenLand v1.2 was examined by two groups of 3 evaluators each. All 6 evaluators had similar (medium) experience in heuristic evaluations (with Nielsen's heuristics), but no experience in usability evaluation of Grid

Computing applications. They all had comparable (low, if some) experience in using Grid Computing applications.

As in the previous case study, the first group performed a heuristic evaluation using only the 12 new (Grid Computing) heuristic (based on the full heuristics' specification). The second group performed a similar heuristic evaluation, but using only Nielsen's 10 heuristics. Table 3 shows the number of usability problems identify by each group of evaluators.

TABLE III. NUMBER OF USABILITY PROBLEMS IDENTIFIED IN *GREENVIEW*, BY HEURISTICS

Group 1: Using Grid Computing Heuristics		**Group 2: Using Nielsen's Heuristics**	
ID	***Number of problems***	***ID***	***Number of problems***
H1	4	N2	6
H2	2		
H3	2	N8	4
H4	5	N1	2
H5	5	N4	5
H6	0	N7	1
H7	0	N6	0
H8	2	N3	2
H9	3		
H10	0	N5	0
H11	1	N9	1
H12	2	N10	1
Total:	**26**	**Total:**	**22**

As in the previous case study, more usability problems were captured using Grid Computing heuristics than using Nielsen's heuristics. Analyzing the total of 48 problems identified by the 6 evaluators, we found that:

- 14 problems (29%) were identified by both groups of evaluators,
- 22 problems (46%) were identified only by the group which used Grid Computing heuristics,
- 12 problems (25%) were identified only by the group which used Nielsen's heuristics.

The results proved once again that Grid Computing heuristics work better than Nielsen's heuristics. Problems identified only by Nielsen's heuristics were in fact qualified as minor: only 2 of 12 problems had an average severity over 2, in a five points scale. Most of them were also discharged by a usability test, performed with 4 users.

As in the previous case study, most of the 22 problems identified only by Grid Computing heuristics were qualified as sever. Most of them had an average severity of 2.5 or superior, on a five point scale. Moreover, 1 problem had an average severity of 4, and 3 problems had average severities of 3.67!

C. *Grid Computing Applications as Intercultural Collaboration Platform*

Grid Computing applications are usually intercultural collaboration platforms, and heuristic evaluations of both *GreenView* and *GreenLand* were cross-cultural challenges. Two European projects were evaluated using the 12 Grid Computing heuristics proposed by a Latin American (Chilean) team. Heuristic evaluators were also Chileans. There were both cultural (Latin American vs. European) and language (English vs. Spanish) barriers.

No significant culture–related problems where highlighted during the heuristic evaluations performed on both *GreenView* and *GreenLand*. The 12 usability heuristics proved to be effective tools when evaluating Grid Computing applications as intercultural collaboration platforms [15].

V. CONCLUSIONS

Grid Computing has nowadays a wide range of applications. Even if the current use of Grid Computing is at the hand of experts and researchers with extensive knowledge, it is expected that in the future the technical knowledge of grid users will decrease. That is why we think the usability of Grid Computing applications will soon become a main issue.

Research usually focuses on Grid Computing based application development from a technical point of view; there is a need for new evaluation methods or at least usability evaluations should be particularized for Grid Computing environments.

A set of 12 specific usability heuristics for Grid Computing applications was proposed. The new heuristics were validated through two case studies. Their potential was also checked in usability evaluations of Grid Computing applications as intercultural collaboration platforms.

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] R. Otaiza, C. Rusu, and S. Roncagliolo, "Evaluating the Usability of Transactional Web Sites", Proc. 3rd International Conferences on Advances in Computer-Human Interactions (ACHI 2010), IEEE Press, pp. 32-37, ISBN-13: 978-0-7695-3957-7, 2010.

[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, pp. 324-328, ISBN: 978-1-4244-3351-3, 2009.

[3] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International J. Supercomputer Applications, vol. 15(3), pp. 200-222, 2001.

[4] J. Linderoth, "An Introduction to the Computational Grid", University of Wisconsin, 2007.

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

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

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

[8] T. Doyle, "GridPP: the UK's contribution to the international collaboration building a worldwide Grid, the LHC Computing Grid: is the system usable?" University of Glasgow, 2006.

[9] B. Hagemeier and M. Kozlovszky, "TG9 – Grid Portals and User Interfaces", Research Center Juelich, 2008.

[10] A. Walker, T. White, R. Bruin, and K. Austen, "Usable grid infrastructures: practical experiences from the eMinerals project", University of Cambridge, 2008.

[11] 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), IEEE Press (accepted paper).

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

[13] J. Nielsen, "Ten Usability Heuristics", Available at www.useit.com/papers/heuristic/heuristic_list.html (last access 12/10/2010).

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

[15] 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, pp. 179-182, ACM 978-1-4503-0108-4/10/08.

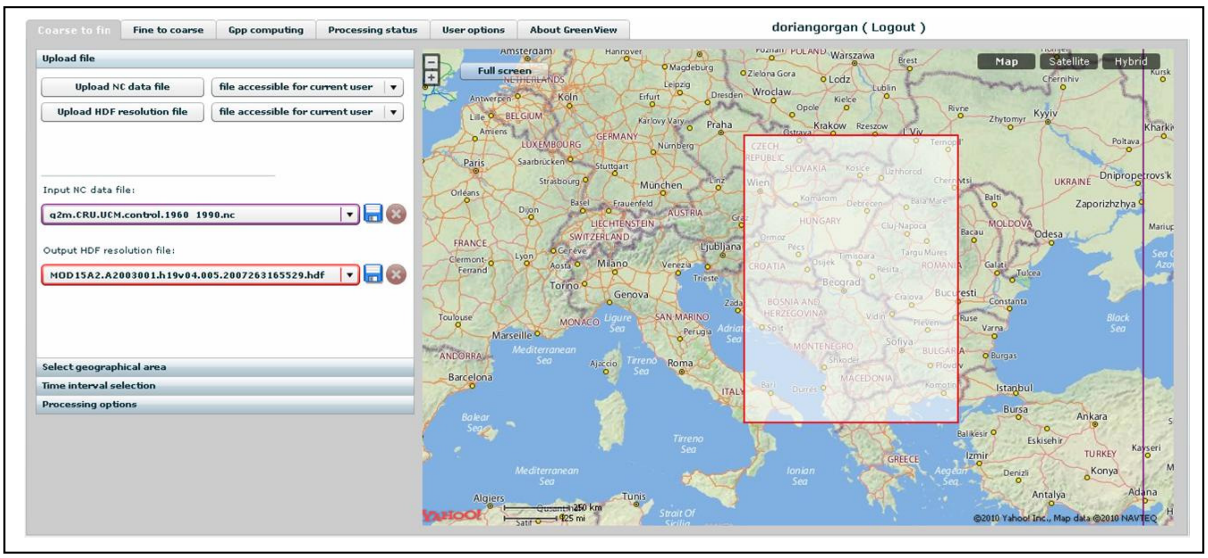

Figure 1. *GreenView* application shotscreen.

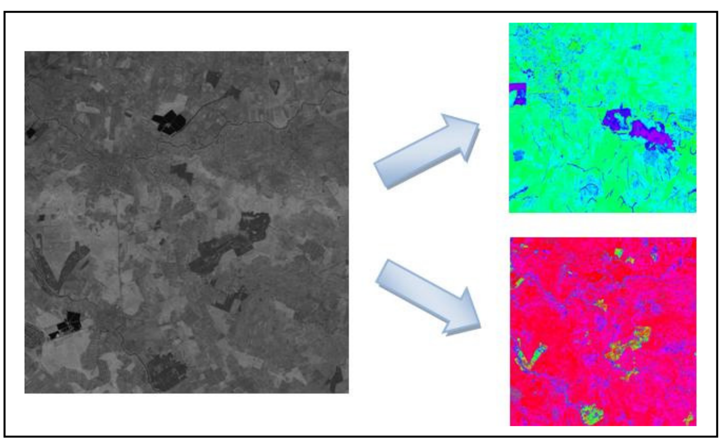

Figure 2. *GreenLand* thematic maps processes.