

# First Experimentation of the ErgoPNets Method Using Dynamic Modeling to Communicate Usability Evaluation Results

Stéphanie Bernonville<sup>1,2,3,4</sup>, Christophe Kolski<sup>1,2,3</sup>, Nicolas Leroy<sup>4</sup>,  
and Marie-Catherine Beuscart-Zéphir<sup>4</sup>

<sup>1</sup> Univ Lille Nord de France, F-59000 Lille, France

<sup>2</sup> UVHC, LAMIH, F-59313 Valenciennes, France

<sup>3</sup> CNRS, UMR 8530, F-59313 Valenciennes, France

<sup>4</sup> EVALAB-EA 2694, Faculté de Médecine, 1 place de Verdun, F-59045 Lille, France  
{sbernonville,nicolas.leroy-2,mcbeuscart}@univ-lille2.fr,  
christophe.kolski@univ-valenciennes.fr

**Abstract.** When a computer application is being designed or re-engineered, especially a user-centred application, communication between ergonomists and computer scientists is very important. However, the formalisms used to describe ergonomic problems and recommendations are often based on natural language. Consequently, the results of ergonomic evaluation can be poorly understood or interpreted by computer scientists. To remedy this problem, we propose a method, called ErgoPNets. The method creates common work support for both the ergonomists and the computer scientists working on the same project. Comprehensible for everyone, this support must provide an efficient tool that can be used by each person involved. ErgoPNets uses Petri nets to model Human-Computer Interaction (HCI) procedures and ergonomic criteria to model the ergonomic analysis. A first experimentation has been performed with designers/developers and academic researchers.

**Keywords:** Human-Computer Interaction (HCI), HCI modelling, usability problems, ergonomic criteria, petri nets, ErgoPNets Method, critical system.

## 1 Introduction

In a design or re-engineering project, the role of ergonomists is to evaluate with their own methods (e.g., ergonomic evaluations and/or user tests) mocks-up, prototypes or applications where some are more complex than others (i.e. healthcare domain). In this case, it is necessary to describe in a rigorous way the detected problems and recommendations to avoid computer system failure or malfunctions [1], [2], [3].

To proceed to the ergonomic evaluation, ergonomists can analyse user activity and/or base their evaluations, recommendations and justifications of their results on the experience gained during previous analyses. Obviously, the ergonomists and the computer scientists must work in close collaboration to successfully complete the project [4].

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-3-642-11750-3\\_10](https://doi.org/10.1007/978-3-642-11750-3_10)

P. Palanque, J. Vanderdonckt, and M. Winckler (Eds.): HESSD 2009, LNCS 5962, pp. 81–95, 2010.

© Springer-Verlag Berlin Heidelberg 2010

Following an activity analysis or an ergonomic evaluation, ergonomists often encounter problems in communicating their ideas, recommendations and results [5], especially when the information must be interpreted by the computer scientists. In this study, we concentrate on the communication of the results from an ergonomic evaluation of computer tools and present an example to illustrate this problem.

Ergonomists perform usability inspections—such as cognitive inspections (e.g., cognitive walkthrough) [6], evaluations of conformity to recommendations (e.g., guideline reviews) [7], evaluations of conformity to ergonomic dimensions (e.g., standards inspections) [8]—in order to detect ergonomic problems and then recommend actions to take to solve the problems. Figure 1 shows one type of form which gives problems and recommendations from a project on which several of the authors of the present article worked. This type of formalisation is practically always accompanied by a screenshot of the software display and, when necessary, by the ergonomists' model of the solution to the problem detected. The chart includes a “criteria” column showing the ergonomic criterion that corresponds to the problem detected, a “problem description” column containing a text explanation of the problem, a “consequences” column indicating the possible risks that the problem could cause, a “recommendations” column containing a text explanation of the ergonomists’ suggestions and a “degree of gravity” column indicating the seriousness of the problem, which could range from one star (not serious) to four stars (extremely serious). This type of description is simple to read but can lead to comprehension and interpretation problems. The fact that the problem detected and the recommendation for solving it are both described in natural language can give rise to the following problems: the problem description may be ambiguous, forcing the computer scientist to read the text several times in order to understand the problem, and even then he/she might misinterpret the description; the recommendation may not provide a precise solution; or the problem and/or its solution may not be situated in terms of the overall system dynamics.

Criteria	Problem Description	Consequences	Recommendations	Degree of gravity
<i>Consistency</i>	Certain symbols do not mean the same thing on different screen pages; for instance: “A” for the stopping of an order and “A” for an absence of patient progress, “R” for reactivate and renew.	Risk of error/disturbance Increase in memory workload	Do not use the same symbol or word to mean different things; always use the same formulation for the same meaning.	★★

**Fig. 1.** Example of a form used by ergonomists for reporting ergonomic problems and recommendations, (translation of a form given in [9])

This article proposes a possible solution for solving these problems of comprehension and interpretation by suggesting a precise location of the problems detected and recommendations made, directly onto procedures. Moreover, the ErgoPNets method, works to create a common work support [10], [11], [12] [13] [14]. This method makes it possible to model the ergonomic problems detected by the ergonomists, as well as their recommendations. ErgoPNets models incorporate the

formal modelling method, *Petri nets* and the ergonomic criteria that are used in usability engineering [15], [16].

The first section presents the ErgoPNets method. The second section describes the experimentation of the method and provides the first results obtained. The last section offers our conclusion and our perspectives for research.

## 2 Presentation of the ErgoPNets Method

In this section, we present a detailed description of the ErgoPNets method, including both the Petri nets and the ergonomic criteria. We discuss the reasons which led us to choose the Petri nets formalism and the ergonomic criteria.

### 2.1 Petri Nets

Petri nets (PN) have been used in Human-Computer Interaction (HCI) for almost twenty years. They were first used to model human tasks [17] and [18], then they progressively came to be used in the specification and design of targeted interactive systems, particularly dynamic systems [19] [20] [21]; see for example articles about ICO (Interactive Cooperative Objects): [22], [18] and [23]. When linked to the object concept, Petri nets are used as a modelling tool in Task Object Oriented Design (TOOD) [24] [25], which aims to provide a method that covers the entire design process from task modelling to HCI parts generation. Palanque and his colleagues have proposed rule-based mechanisms for the automatic evaluation of PN-based models of interactive systems [26]. Ezzedine and Kolski [27] used Petri nets to study the functional tasks of a technical system in normal and abnormal situations, in order to facilitate the specification of interactive systems. Petri nets can also be used as a tool for formal comparisons of a set human task (theory) and the corresponding real task (practice) [28]. These diverse use possibilities make Petri nets a good choice for our application. We also chose to use Petri nets because, in addition to their varied uses, they make it possible to represent the task's dynamic dimension graphically. In our context, they also make it possible to model the procedures provided by the software and to associate these procedures to the ergonomists' recommendations.

### 2.2 The Ergonomic Criteria

Ergonomic inspection is commonly used to judge the conformity of computer interfaces to usability principles [29]. During such inspections, a small group of ergonomists examine the interfaces in detail in order to assess their conformity. Though some ergonomists base their judgements on experience and intuition alone, the application of certain basic rules, set out in the form of guidelines, is recommended [30]. Amongst the most commonly-used rule structures for ergonomic inspections, we chose one taken from the ergonomic criteria developed by Bastien and Scapin [31]. To develop these criteria, experimental results and recommendations were synthesized and translated into rules, which were then grouped together, creating 8 criteria and 13 sub-criteria [31] (Fig. 2). The results of an experimental study project by Bastien and Scapin [32] showed that these ergonomic criteria were more efficient than the dialogue standards ISO/DIS 9241-10 with respect to detecting ergonomic problems on user interfaces.

Main Criteria	Sub-criteria
1. Guidance	1.1 Prompting 1.2 Grouping/Distinction by location 1.3 Immediate feedback 1.4 Legibility
2. Workload	2.1 Brevity 2.2 Information density
3. Explicit control	3.1 Explicit user action 3.2 User control
4. Adaptability	4.1 Flexibility 4.2 User experience
5. Error Management	5.1 Error protection 5.2 Quality of error messages 5.3 Error correction
6. Consistency	
7. Significance of codes	
8. Compatibility	

**Fig. 2.** Classification of the ergonomic criteria and sub-criteria developed by Bastien and Scapin

The criteria described above were created to help HCI evaluators detect problems during ergonomic inspections; they also represent the main ergonomic dimensions according to which an interactive software programme may be specified or evaluated. We therefore use them to categorize problems detected using other methods, such as observation or user testing. Indeed, the criteria were specifically designed in order to be used by both human factors specialists and non-specialists [32].

### 2.3 Principles of the ErgoPNets Method

**Basic stages of the ErgoPNets method.** The ErgoPNets method has five stages which can be applied after an ergonomic evaluation performed by ergonomists: context definition, current procedure description, problem identification and explanation, recommended procedure description, and recommendation identification and explanation.









1. *context definition*: Defining the context means characterizing the software analysed (which could mean providing a specific reference to a report mentioning the software's HCI specification) and identifying the user's objective. This helps to clarify the context.
2. *current procedure description*: In this stage, the procedure provided by the existing software, prototype or mock-up, and corresponding to the user objective identified in stage 1, is described with the help of Petri nets, indicating the user actions and the results of these user actions with the current software.
3. *problem identification and explanation*: In this stage, the ergonomic problem is identified and its specific place in the procedure described in stage 2 is situated. Textual explanations are provided to clarify the extent and possible consequences of the problem.

4. *recommended procedure description*: In this stage, Petri nets are also used to describe a procedure that integrates the ergonomists' recommendation, and perhaps to provide a mock-up of the recommended procedure. This new procedure may be completely different from the one described in stage 2 or may just be changed slightly.
5. *recommendation localization and explanation*: In this stage, the changes recommended by the ergonomist are situated in the procedure described in stage 4 and some textual explanations are provided to clarify the improvement brought about by the changes.

Stages two and four call for an adapted formalization using Petri Nets. System states or places (small circles) can be actions taken by the user or by the computer application, and transitions (small rectangles) are the events that allow movement from one state to another. The places and transitions are linked by arcs (arrows). Each place and each transition is described in words, using “and”, “or” and “then”, thus allowing several actions or several events to be represented. These words imply different things. For example, using the word “and” does not impose a specific order of actions/events; however, the word “then” does.

Stages three and five identify and locate the sets of places and transitions that correspond to the problem and the recommendation. The locations are represented with a dotted line framing the part of the procedure in which the problem/recommendation is found. In stage three, the ergonomic criteria—represented by icons and text describing the problems—are applied. Each criterion (issued from the work of Bastien & Scapin) has its own icon (shown in Table 1).

**Table 1.** Icons that represent the ergonomic criteria



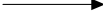



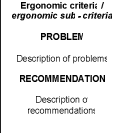




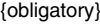
Ergonomic criteria	Icons	Icons explanation
Guidance		Image of a signpost indicating directions to show persons the way
Workload		Image of the brain allowing the human-being to work
Explicit control		Image of remote control buttons (reverse and forward) allowing an explicit control of a video recorder
Adaptability		Image of the belt adjustable to the waist of persons
Error management		Image of a warning notice board to prevent danger
Consistency		Group of non consistency forms
Significance of codes		Question mark meaning incomprehension
compatibility		Image of pieces of puzzle can be assembled

In stage five, the sign “R” (for recommendation) is used to indicate the result of a recommendation. The text that describes the problems and the recommendations is located in a text zone between two procedures and includes the name of the criteria and, maybe, the sub-criteria. This text zone is linked to the icons by a black line. It is also possible to show the correspondence between the two procedures with dotted grey lines, which allows the differences between two procedures to be highlighted.

The amount of detail in the procedures has been adapted to each situation. For example, some events were simplified because they are not necessary to understand the ergonomic problems (asterisk: \*). However, these simplifications are still shown in order to represent a complete and logical procedure that fulfils the initial objective. To represent the information that *must* be given in interactive systems, the word "obligatory" can be added between brackets. Finally, it is possible to add comments about any element of the model using a rectangle with a turned down corner.

The procedure models provided by the software and the procedures illustrated by the mock-ups provide a good support that could facilitate software design and re-engineering. Ergonomists can use them to represent their recommendations in a manner that can be more easily exploited by development teams. The graphic representations used in the ErgoPNets method are shown in Table 2.

**Table 2.** Different graphic representations used in ErgoPNets

Forms	Meaning
	Place (system state)
	Transition (event)
	Link between the state of the system and the event
	Dotted line used to frame the problems and recommendations on the Petri Net
	Correspondence between procedures
	Sign indicating a proposed recommendation (place on the criterion icon to identify recommendations on the Petri Net)
	Text zone describing problems and recommendations
	Link associating an icon with a text zone
	Simplified procedure
	The use of the words « and », « or » and « then » allows several actions or events to be represented. The word «and » does not impose an order, but the word « then » does.
	Note or comment
	Constraint indicating a compulsory event in the procedure

## 2.4 Tool Supporting the ErgoPNets Method

At the moment, the ErgoPNets models are built using the Visio© software. This software makes it possible to create plans and diagrams starting from predefined shapes. These shapes are organised in "template" files and are classified according to categories (e.g., software, flux diagram, network). To make a diagram, users simply need to choose the template they want and drag the shapes onto the drawing. Visio© makes it possible for users to create new templates and to define their own shapes. Using this software, we created an ErgoPNets template containing the various shapes needed to create a model. (All of these shapes have been shown above in Tables 1 and 2.) These are the graphic elements that allow Petri nets to be created and ergonomic problems and recommendations to be identified and described. Figure 3 shows an example of an ErgoPNets model created with the Visio© software. We are currently developing a software tool specifically for the ErgoPNets method. It is intended for use by both ergonomists and computer scientists. This software tool will allow the creation of common work supports to facilitate the modelling and analysis of the ergonomic problems detected, as well as the recommendations for solving these problems. In this version, this tool will not support the simulation of the models (see for instance PetShop supporting the simulation of ICOs [33]).

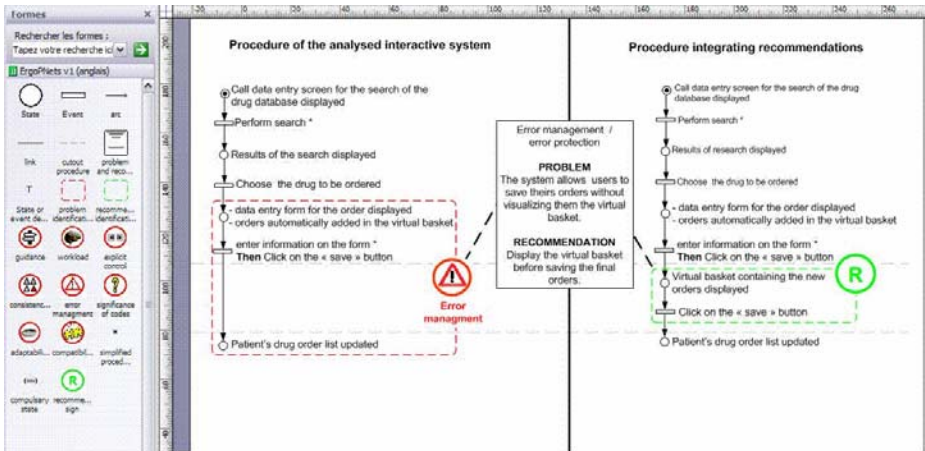


Fig. 3. Example of an ErgoPNets model created with the Visio© software

## 3 Experimentation of the ErgoPNets Method

The objective of the experimentation was to verify the following hypothesis: (1) a best comprehension level of problems and recommendations must be obtained with the ErgoPNets method compared with text based formalism, (2) the ErgoPNets is more relevant for problems linked to dynamic procedures.

For this, we asked a group of participants to estimate the comprehension of problems and recommendations described with the ErgoPNets method and with a

based text formalism (an example of “table” formalism is described in figure 1) by comparing the final results of each method. The testing involved first reading a set of models described with the ErgoPNets method and then a second set of models described with the “table” formalism. This type of experimentation had the further advantage of allowing us to gather participant opinions about the ErgoPNets method.

### 3.1 Participants

Twelve participants were involved. Six designers/developers came from development companies who had worked on interactive software design projects. Their average age was 35 years, and they all had at least 5 years experience. This group was made up of people who used to work with ergonomists and therefore had analysed the descriptions of ergonomic problems and recommendations provided by the ergonomists. The second group included six academic researchers, specialists in software engineering and Human-Computer Interaction. Their average age was 30 years, and they all had at least 2 years experience. These people had a high level of knowledge about software design projects and about software engineering and HCI methods and models.

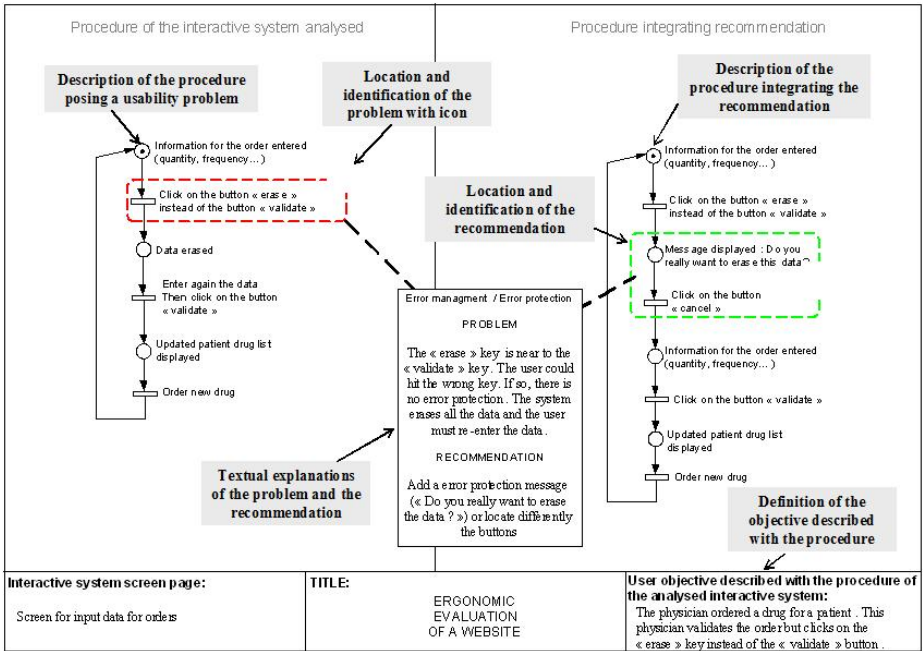
The objective was to test their understanding of the problems and recommendations described with the ErgoPNets method and the “table” formalism. The different nature of the two groups allowed us to collect information about ErgoPNets from two different points of view: (1) potential users and (2) methods and models specialists (i.e., software engineering researchers).

### 3.2 Procedure

At the beginning of each test some explanation and instruction was given to the participants as the experimentation objectives, the important concepts of the two evaluated methods, the CPOE software used in the test and finally the different steps of the test. Twelve models (2 sets of 6 models) were tested. These models cover six potentially critical ergonomic problems, with the corresponding recommendations to solve them. The types of problems dealt with are shown in figure 6 (e.g., Guidance/incitation, error management/ quality of error messages). They come from a real ergonomic evaluation of an existing CPOE (Computer Physician Order Entry) software used in hospitals. The testing did not focus on the completeness of the ergonomic criteria; indeed, the problems were chosen, based on typical problems associated to the procedures provided by the software. Examples of support used during the test are presented in fig. 4 and fig. 5.

Two groups have been planned for the evaluation. Each group was made up of three designers and three academic researchers. The first group tested problems numbered 1, 2 and 3 (see Fig. 6) described with the ErgoPNets method as well as problems numbered 4, 5 and 6 (see Fig. 6) described with “table” formalism. While the second group tested problems numbered 1, 2 and 3 (see Fig. 6) described with the “table” formalism as well as problems numbered 4, 5 and 6 (see Fig. 6) described with the ErgoPNets method. This organisation allowed facilitating the comparison of problems and their recommendations with the different participants.





**Fig. 4.** Example of a model (n°5), described with the ErgoPNets method, involving the same problem and the same recommendation as the model in Figure 5 (please note that grey boxes are just comments for reader of this paper)

Criteria	Problem description	Consequences	Recommendations	Degree of gravity
Error management/ error protection	The « erase » key is near to the « validate » key. The user could hit the wrong key. If so, there is no error protection. The system erases all the data and the user must re-enter the data.	Time loss	Add a error protection message (« Do you really want to erase the data? ») or locate the keys differently	☆☆

**Fig. 5.** Example of a model (n°5), described with the “table” formalism

Six questionnaires were given to the participants, who were asked to evaluate each model read. These questionnaires allowed us to collect data about the comprehension of the problems and recommendations described with each formalism: ErgoPNets or the “table” formalism. The participants must estimate their level of comprehension on a scale from 0 (complicated) to 10 (very clear). Finally, a global evaluation questionnaire was filled out, giving the opinion of the participants on the ErgoPNets method (difficulties encountered, deficiencies in the problem/recommendation descriptions, overall satisfaction/dissatisfaction).







Problems	ErgoPNets models	« Table » models
 1. Guidance/incitation	GROUP 1	GROUP 2
 2. Error management/ quality of error messages		
 3. Error management/ error protection		
 4. Adaptability/flexibility	GROUP 2	GROUP 1
 5. Error management/ error protection		
 6. Guidance/incitation		

Fig. 6. Organisation of models within the experimental protocol

### 3.3 Results

**Comparison of results concerning the comprehension of problems and recommendations with the ErgoPNets method and the "table" formalism.** To obtain the results presented below, we asked the participants their opinion about models of problem and models of recommendation. These indications were then compared to obtain information about the level of comprehension for 6 problems and their corresponding recommendations when using the ErgoPNets method and the "table" formalism. In general, the average scores obtained for problems modeled with the ErgoPNets method were superior to those obtained problems modeled with the "table" formalism, except for problems n°4 and n°6 (fig. 7, on the left).

For problem n°4, the software imposed a keyboarding order that was different from the usual physician ordering activity. The majority of the participants found that a textual explanation of the problem was sufficiently clear and that the ErgoPNets method was not necessary. They said they didn't need to understand the consequences of an action in a procedure.

For problem n°6, there was also a keyboarding problem. A textbox on a screen prompted the user to input data into this textbox which the software did not permit. Therefore, for the same reasons as in problem n°4, the majority of the participants found that a textual explanation was sufficient to describe the problem.

Our results also show that comprehending the problems and the recommendations was not easy for everyone, whatever the method used to describe the problem. Indeed, as the important differences on the graph in fig. 7 (on the left) show, the participants are divided. For example, for problem n°3 described with the ErgoPNets method, the scores were between 0 and 9.8. Two out of six participants gave a score between 0 and 5 (i.e., between *complicated* and *moderately clear*), while four out of six participants gave a score between 5 and 10 (i.e., between *moderately clear* and *very clear*). For problem n°3 described with the « table » formalism, the scores were positioned between 2 and 7.3. Four out of six participants gave a score between 0 and 5 (i.e., between *complicated* and *moderately clear*) and two out of six participants

gave a score between 5 and 10 (between *moderately clear* and *very clear*). Thus, problem 3 was considered to be described more clearly with the ErgoPNets method, but the gap between the participant scores shows that these results are not unanimous.

The results for the recommendations were similar to those obtained for the problems. Globally, the recommendations were better understood if described with the ErgoPNets method, given the average scores obtained with the ErgoPNets method were superior to those obtained with the “table” formalism (fig. 7, on the right). The only exception was recommendation n°4, for which several participants felt that a simple textual description could be enough. Nonetheless, the differences between participants are important, underlining that the results do not show a clear advantage for ErgoPNets and those opinions are divided on the subject. In general, the participants liked the complete descriptions of the recommendations because, in their experience, the ergonomists' recommendations are often vague and open to interpretation.

In conclusion, these results are encouraging, but improvements still have to be made to obtain a clear mandate for ErgoPNets. The test has also helped to point out that the ErgoPNets method may be more effective for describing certain types of problems, especially for those associated with the procedure. The next section completes these results with a synthesis of all the information collected through the participant comments about the ErgoPNets method (e.g., participant satisfaction/dissatisfaction, lack of elements in the method proposed...).

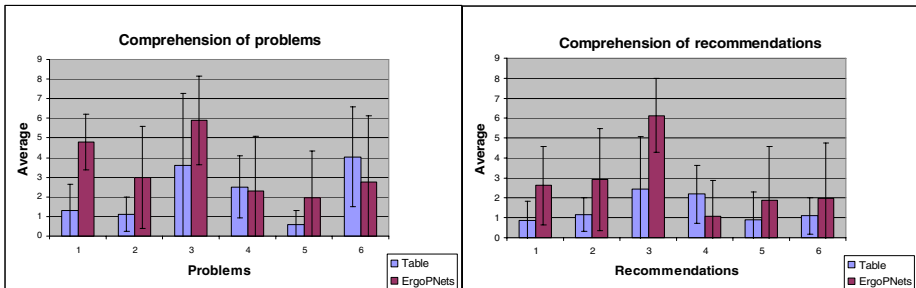


Fig. 7. Results obtained for the problems and the recommendations

**Synthesis of the results based on the responses to global questionnaire and the comments recorded during the test.** The global questionnaire and the comments recorded during the test allowed us to collect participant opinions about the ErgoPNets method. The questionnaire included questions about participant satisfaction, the difficulties encountered and the deficiencies of the ErgoPNets method. The comments of the designers and the developers were particularly useful, since they were able to assess the method's utility for their professional activity.

Table 3 reproduces some of comments recorded during the test. These comments were sorted into (1) positive points and the advantages of ErgoPNets and (2) negative points and disadvantages of the method.

To measure participant satisfaction, we asked participants to indicate their level of satisfaction on a scale of 0 (not at all satisfied) to 10 (totally satisfied). The average

score was 8.1 for the computer scientists, with a standard deviation equal to 1.60, and 7.8 for the researchers, with a standard deviation equal to 1.80. These results indicate a fairly positive view of the ErgoPNets method. Furthermore, five out of six computer-scientists answered *yes* to the question "Could the method ErgoPNets be helpful in your professional activity?".

In conclusion, all information collected allowed us, first, to assess the potential contribution of the ErgoPNets method for ergonomic evaluations of software within design or re-engineering projects. This method offers a promising way to describe the problems and recommendations encountered. In addition, the test also allowed us to identify the improvements that can be made (such as: addition of the degree of gravity). All of these remarks have been taken into account, and a new version of the ErgoPNets method is under development. When it is ready, a new test will be planned to evaluate the method's evolution.

**Table 3.** Synthesis of the comments recorded during the test (The same comment attributed to more than one subject means that their comments were very close.) (CS= Computer Scientist, R= Researcher).

Participants concerned	Positive points and advantages of the ErgoPNets method	Participants concerned	Negative points and disadvantages of the ErgoPNets method
CS2, CS6, CS7	Exhaustive and obvious representation of software procedures	CS5, CS6	It's still only « gymnastics for the mind »
CS2, CS7	The method allows a comparison between the procedure integrating the problem and the procedure integrating the recommendation	R3, CS2	It could weigh down simple cases
R1	« The description of the procedures show one coherent result with all the project participants »	R1, CS7	Here the page is overloaded
CS2	«This type of method is less subject to interpretation»	CS6, CS7, R6	The model takes a long time to read
CS5, R1	This method helps to clarify things	CS7, R1	It can be complicated when there are several options for recommendations
CS6	«I find it easy because there is a logical sequence of actions»	R3	«However the method is not suitable for all that is only visual»
CS2, CS5	The model is enough to help developers working all alone	CS4	«Here I understood the ergonomic problem only after reading the recommendation»
R3	«ErgoPNets is appropriate for anything interactive»		
CS4	«Nothing better than diagrams to explain things»		
CS4	«We can see the different stages, It is better than a textual description where there is too much talk»		
CS5, CS7	The description allows the developer to see the result of a change in the procedure, it argues for the modification		

## 4 Conclusion and Perspectives

In this article, we presented a first experimentation concerning the ErgoPNets method helping to describe ergonomic problems and their recommendations. Its objective is to propose a formalism more rigorous than text-based description used currently in evaluation which can engender comprehension and interpretation problems. The ErgoPNets method combines Petri Nets and ergonomic criteria. This combination makes it possible to take two important aspects into account: (1) procedure descriptions and prescriptions and (2) HCI evaluation results, given the recommendations to be considered. The results of our tests of the ErgoPNets method

show the method's potential for use in the evaluation phase of a design or re-engineering project. Even if they suggest further investigation, these results have allowed us to continue to improve certain aspects of the method. For example, we were able to improve the graphic representation in ErgoPNets by adding relevant elements and adapting the Petri nets to make the models easily understandable.

The ErgoPNets method is increasingly being used for real projects by ergonomists at the EVALAB Laboratory (see <http://www3.univ-lille2.fr/evalab/>).

For the moment, the tool supporting ErgoPNets is in the form of a Visio© template. We intend to develop this tool further, more specifically by allowing the verification of the Petri net properties. Such property verifications (e.g., network limits, network vivacity, network reversibility, or network blockage) will insure the formal character of the Petri nets created by ErgoPNets users and remove any potential incoherencies. At the moment, a first application has been developed. It takes into account the verification of rules linked to Petri net formalism. Our long term research perspective is to assemble, create and adapt software engineering methods and models in order to develop a multi-model approach that will facilitate the communication between the various partners in an information system design or re-engineering project.

**Acknowledgements.** The authors wish to thank the ergonomists and the computer scientists who participated in this study. They would also like to thank the Nord/Pas-de-Calais Regional Council, the CHRU of Lille, the FEDER, the RNTS network, and the French Ministry of Educ., Research and Technologies for supporting this research.

## References

1. Horsky, J., Kuperman, G.J., Patel, V.L.: Comprehensive analysis of medication dosing error related to CPOE. *JAMIA* 12, 337–382 (2005)
2. Leveson, N.G., Turner, C.S.: An investigation of the Therac-25 accidents. *Computer* 26, 18–41 (1993)
3. Taylor, J.R.: The contribution of the design to accidents. *Safety Sciences* 45, 61–73 (2007)
4. Livari, N.: Representing the User' in software development—a cultural analysis of usability work in the product development context. *Inter. With Computers* 18, 635–664 (2006)
5. Gulliksen, J., Boivie, I., Göransson, B.: Usability professionals—current practices and future development. *Interacting With Computers* 18, 568–600 (2006)
6. Lewis, C., Wharton, C.: Cognitive Walkthroughs. In: Helander, M., Landauer, T.K. (eds.) *Handbook of Human-Computer Interaction*, 2nd edn., pp. 717–732. Elsevier Science, Amsterdam (1997)
7. Microsoft: *The Windows Interface Guidelines for Software Design: An application Design Guide*. Microsoft Press, Redmond (1995)
8. Shneiderman, B.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd edn. Addison-Wesley Publishing Company, Massachusetts (1992)
9. Evalab, Ergonomic evaluation of the GENOIS software, Project Report, Evalab Laboratory Lille, France (2006)

10. Bernonville, S., Kolski, C., Beuscart-Zéphir, M.C.: Contribution and limits of UML models for task modelling in a complex organizational context: case study in the healthcare domain. In: Soliman, K.S. (ed.) *Internet and Information Technology in Modern Organizations: Challenges & Answers*, Proc. 5th IBIMA Conf., Cairo, Egypt, pp. 119–127 (2005)
11. Bernonville, S., Kolski, C., Beuscart-Zéphir, M.C.: Towards an Assistance for Selecting Methods and Models for Interactive Software Design or Re-engineering within Complex Organisation: Application Case of a CPOE Software. In: Badr, Y., Chbeir, R., Pichappan, P. (eds.) *Proc. of the second IEEE Int. Conference on Digital Information Management*, Lyon, Workshop sessions, October 28–31, pp. 597–602. IEEE Press, Los Alamitos (2007)
12. Bernonville, S., Leroy, N., Kolski, C., Beuscart-Zéphir, M.C.: Explicit combination between Petri Nets and ergonomic criteria: basic principles of the ErgoPNets method. In: *Proc. of the 25th Edition of EAM 2006, European Annual Conference on Human Decision-Making and Manual Control*, Valenciennes, PUV, September 27–29 (2006) ISBN 2-905725-87-7
13. Beuscart-Zéphir, M.C., Pelayo, S., Guerlinger, S., Anceaux, F., Kulik, J.F., Meaux, J.J., Degoulet, P.: Computerized “Physician” Order Entry (CPOE): missing the “N”, standing for Nurse. In: *Physicians and Nurses activity analysis and comparison with Paper-based and Computerized Order Entry systems*, IT in Health Care: Sociotechnical Approaches, Portland Oregon, USA (2004)
14. Bernonville, S., Kolski, C., Leroy, N., Beuscart-Zéphir, M.: Integrating the SE and HCI models in the human factors engineering cycle for re-engineering Computerized Physician Order Entry systems for medications: basic principles illustrated by a case study. *International Journal of Medical Informatics*, doi:10.1016/j.ijmedinf.2008.04.003 (in press)
15. Mayhew, D.J.: *The usability engineering lifecycle*. Morgan Kaufmann Publishers, San Francisco (1999)
16. Cockton, G., Lavery, D., Woolrych, A.: Inspection-based evaluations. In: Jacko, J.A., Sears, A. (eds.) *Handbook of Task Analysis for Human-Computer Interaction*, pp. 1118–1138. Lawrence Erlbaum Associates, London (2003)
17. Abed, M., Ezzedine, H.: Vers une démarche intégrée de conception-évaluation des systèmes Homme-Machine. *Journal of Decision Systems* 7, 147–175 (1998)
18. Palanque, P., Bastide, R.: Synergistic modelling of tasks, system and users using formal specification techniques. *Interacting With Computers* 9, 129–153 (1997)
19. Moussa, F., Riahi, M., Kolski, C., Moalla, M.: Interpreted Petri Nets used for human-machine dialogue specification. *Integrated Computer-Aided Engineering* 9, 87–98 (2002)
20. De Rosis, F., Pizzutilo, S., De Carolis, B.: Formal description and evaluation of user-adapted interfaces. *International Journal of Human-Computer Studies* 49, 95–120 (1998)
21. Ezzedine, H., Trabelsi, A., Kolski, C.: Modelling of an interactive system with an agent-based architecture using Petri nets, application of the method to the supervision of a transport system. *Mathematics and Computers in Simulation* 70, 358–376 (2006)
22. Palanque, P.: *Modélisation par objets coopératifs interactifs d’interfaces homme-machines dirigées par l’utilisateur*. Thèse de Doctorat, Université de Toulouse 1 (1992)
23. Bastide, R., Navarre, D., Palanque, P.: A tool-supported design framework for safety critical interactive systems. *Interacting with Computers* 15, 309–328 (2003)
24. Tabary, D., Abed, M.: A software environment task object oriented design (ETOOD). *Journal of Systems and Software* 60, 129–140 (2002)
25. Abed, M., Tabary, D., Kolski, C.: Using Formal Specification Techniques for the Modelling of Tasks and Generation of HCI Specifications. In: Diaper, D., Stanton, N. (eds.) *The Handbook of Task Analysis for Human-Computer Interaction*, vol. 5, pp. 503–529. Lawrence Erlbaum Associates, Mahwah (2003)

26. Palanque, P., Farenc, C., Bastide, R.: Embedding Ergonomic Rules As Generic Requirements in a Formal Development Process of Interactive Software. In: Proceedings of Interact 1999, pp. 408–416. IOS Press, Amsterdam (1999)
27. Ezzedine, H., Kolski, C.: Modelling of cognitive activity during normal and abnormal situations using Object Petri Nets, application to a supervision system. *Cognitive, Technology and Work* 7, 167–181 (2005)
28. Abed, M., Bernard, J.M., Angué, J.C.: Method for comparing task model and activity model. In: Proceedings 11th European annual conference Human Decision Making and Manual Control, Valenciennes, France (1992)
29. Kahn, J., Prail, A.: Formal Usability Inspection. In: Nielsen, J., Mack, R.L. (eds.) *Usability inspection method*, pp. 141–171. John Wiley & Son, New York (1993)
30. Vanderdonckt, J., Farenc, C. (eds.): *Tools for working with guidelines*. Springer, London (2000)
31. Bastien, J.M.C., Scapin, D.L.: Ergonomic Criteria for the Evaluation of Human-Computer Interfaces, Rapport technique INRIA, 156 (1993)
32. Bastien, J.M.C., Scapin, D.L., Leulier, C.: The ergonomic criteria and the ISO/DIS 9241-10 dialogue principles: a pilot comparison in an evaluation task. *Interacting With Computers* 11, 299–322 (1999)
33. Barboni, E., Navarre, D., Palanque, P., Bazalgette, D.: PetShop: A Model-Based Tool for the Formal Modelling and Simulation of Interactive Safety Critical Embedded Systems. In: Proceedings of HCI'Aero conference (Demonstration) (HCI'Aero 2006), Seattle, USA (September 2006)