Explicit combination between Petri Nets and ergonomic criteria: basic principles of the ErgoPNets method

Stéphanie Bernonville [1, 2], Nicolas Leroy [2], Christophe Kolski [1], Marie-Catherine Beuscart [2]

(1) LAMIH-UMR 8530, Le Mont Houy, F-50313 Valenciennes cedex 9
stephanie.bernonville@univ-valenciennes.fr

(2) EVALAB-EA 2694, Faculté de Médecine, 1 place de Verdun, F-59045 Lille
mcbeuscart@univ-lille2.fr

Abstract

The method proposed in this article aims to facilitate the re-engineering of existing interactive software by proposing a common framework for Software Engineers and Human Factors specialists. The method explicitly combines Petri Nets and ergonomic criteria. A case study of medication ordering in healthcare is used to illustrate the method.

Keywords: Petri Nets (PN), ergonomic criteria, human-machine interaction, re-engineering

1 Introduction

Ergonomics plays an increasingly important role in interactive software projects. Human factors specialists intervene mainly in human-centred projects, particularly in the requirement analysis, design and evaluation (before and/or after software realization) phases (Mayhew 1999). The goal of such specialists is to help computer scientists develop interactive applications that are adapted to users and their various work situations. To that end, they must communicate results clearly and pertinently. Unfortunately, these different actors—human factors specialists and computer scientists—employ different languages, which can lead to misunderstandings and misinterpretations. To avoid such problems, we propose ErgoPNets, a method that associates Petri Nets (PN) and ergonomic criteria as a means of creating a common framework for computer scientists and human factors specialists.

Petri Nets have already been used in Human-Computer Interaction (HCI) in several application domains to reach a variety of objectives. They have been used to model human tasks and activities (Abed *et al.*, 1992) and to describe interactive system dynamics (cf. ICO: Interactive Cooperative Objects) (Palanque *et al.*, 1997). They have been employed in modeling tools and methods, such as the TOOD (Task Oriented Object Design) method (Tabary, 2002). PN have been used to model normal and abnormal situations involving users (Ezzedine & Kolski, 2005) and to formally compare prescribed tasks and real activities (Abed, 2001). Palanque *et al.* have proposed interesting rule-based mechanisms for the automatic evaluation of PN-based models of interactive systems (Palanque *et al.*, 1999). PN seem to offer interesting possibilities for

HCI, which is why we chose to adapt them for use by human factors specialists and computer scientists.

Like Petri Nets, ergonomic criteria are currently used in the design and evaluation of interactive software (Nielsen & Molich, 1990). The structure proposed by Bastien & Scapin, with groups of 8 criteria and 13 sub-criteria, has become almost standard practice. Based on a synthesis of experimental results and numerous recommendations, (Bastien & Scapin 1993, 1994) designed this structure so that could be used by both designers and evaluators (whether HCI experts or novices).

In the second part of this paper, we explain the overall needs of human factors specialists and computer scientists in terms of communication. The third part describes our ErgoPNets method. The fourth part presents a case study, followed by our conclusion and prospectives for future research.

2 The necessity of better communicating ergonomic data to computer scientists

Cooperation includes communication, coordination and collaboration of a group of actors working to achieve a common objective (Schael, 1997). In such a cooperative context, communication becomes essential. In the situation examined in this paper, the problem is that human factors specialists have difficulties communicating with computer scientists. Indeed, the work methods and the vocabulary used by these two groups are quite different (Bernonville *et al.*, 2005). Ideally, human factors specialists should use paradigms that are understood by computer scientists.

Human factors specialists often have to communicate many different kinds of information (problems, data, concepts or ideas), including evaluation methodologies, scant preliminary or detailed final results, the interpretation of those results, and recommendations for improving the interactive software. Choosing the form for that information can be complicated when the teams are heterogeneous, when the project managers do not have the same needs as the design engineers or the technical writers. As Nayak *et al.* (1995) have observed, "Fitting the content of the communication to the appropriate audience without having to write several flavours of the report is a challenge."

3 ErgoPNets Method

3.1 Basic principles

Petri Nets[1] were chosen in order to describe the dynamic aspect of the task graphically, specifically to model the software procedures provided by designer and the procedures that integrate the recommendations proposed by the human factors specialists. Our

---

[1] Among other factors, Petri Nets allow interactive systems to be specified and designed graphically and dynamically. Other existing software engineering methods, such as UML (Unified Modelling Language), are too general and do not allow the details of interactive system procedures to be model easily. Although the extension Umli (Da Silva and Paton, 2000) seems to provide a support method for UI design, there is no model for describing the links between the interactive actions executed within the system. In the HCI domain, GOMS (Goal Operators Selection rules; Card *et al.*, 1983) allows chains of actions to be modeled; however this method uses text to describe the actions and thus is more difficult to read and interpret than a graphic model.

method takes ergonomic criteria into account, which allows the problems to be categorized ergonomically. Software can then be specified or evaluated according this categorization (Bastien & Scapin, 1994).
Our method has five steps : (1) define the user's objective, (2) describe the procedure provided by the existing software that corresponds to the objective, (3) identify and explain the problems detected by ergonomic analysis, (4) describe the procedure that integrate the human factors specialists' recommendations, which are then illustrated with mock-ups, and (5) identify and explain the recommendations.
Steps two and four call for an adapted formalization of the Petri Nets. System states or places (little circles) can be either actions taken by the user or by the computer application, and transitions (little rectangles) are the events that allow passage from a state to another. Each place and each transition is described in words, with « and », « or » and « then » allowing several actions or several events to be represented. These words are used differently. For example, using the word « and » doesn't impose a specific order or actions/events; however, the word «then » does. Steps three and five apply the ergonomics criteria, which are represented by icons and text describing the problems and the recommendations. Each criterion, as in the work of Bastien & Scapin, corresponds to an icon (table 1). To indicate the result of a recommendation a validation sign is added to the icon. The text that identifies the problems is located in a text zone between two procedures and includes the name of the criteria and, maybe, sub-criteria.

Icons								
Criteria	guidance	workload	Explicit control	adaptability	Error management	consistency	Significance of codes	compatibility

Table 1. Icons that represent the ergonomic criteria.

The amount of detail of the procedures has been adapted to each situation. For example, some events were simplified because they are not necessary to understand the ergonomic problems. However, these simplifications are portrayed so as to represent a complete and logical procedure that fulfils the initial objective.
Models of the procedure provided by the software and the procedure illustrated by the mock-ups provide a good support that could facilitate software design and re-engineering. Human factors specialists can use them to represent their recommendations in a manner that can be more easily exploited by development team.

3.2 Tool supporting ErgoPNets

ErgoPNets Models are realised with VISIO© software. This software allows the creation of organizational diagrams and technical and IT support tools. We have created a library of graphic modelling elements for ErgoPNets. This library includes the icons presented in table 1, forms allowing the creation of Petri Nets, and elements, such as text zones and validation signs. Figure 1 depicts an example application of our method in the VISIO software. To the left of the screen is the library and to the right, the modeling framework.

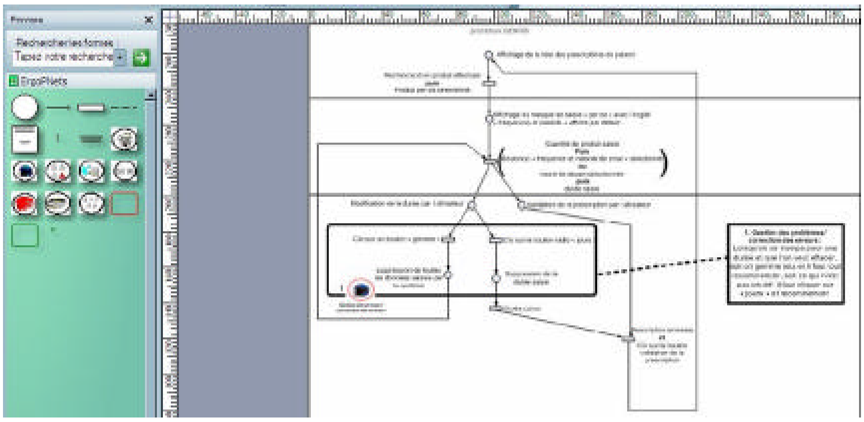

Figure 1.Example of an ErgoPNets model (extracted from Visio).

4 Case study

Ergonomics studies are sometimes incorporated into re-engineering projects, particularly for Web applications and the IT applications used by medical personnel, including physicians, nurses, and pharmacists (Pelayo *et al.*, 2004; Beuscart *et al.*, 2005). Our method was designed in this double context. Figure 2 shows an example of the ErgoPNets modelling support for a medical-related Website. The website proposes a system of physician-patient communication based on secure electronic messaging**.**

The following lines present the five steps of ErgoPNets method as applied to this specific context, highlighting one of the problems detected by ergonomics specialists in terms of website use.

(1) Description of the user's objective: Here, the objective is to read a message and respond to it.

(2) Modeling of the procedure used by the existing software: the procedure that allows the user to read and respond to a message is modelled with the Petri Nets located at the left of figure 2. Places (little circles) are used to describe the actions of the users or the IT application (e.g.,"display the home page"). Transitions (little rectangles) are used to describe the events (e.g.,"click on my electronic messages")

(3) Ergonomic analysis of the procedure: the ergonomic problems found during the usability assessment of the software are modelled. In the example shown in Figure 2, the problem identified has been categorized as being related to the ergonomic criteria classification, "workload". In the Petri Net, the problems corresponding to the set of transitions and states have been identified**.** This set is encircled and associated to the icon, "workload". A text zone within the criteria classification (the sub-criteria "minimal action") proposes an explanation of the problem.

(4) Modeling the integration of the ergonomic specialist's recommendations: After identifying the problems, ergonomic specialists propose a solution. A second Petri Net with the same objective as the first one, but integrating the recommendation in then constructed (Petri Net at the right of figure 2).

To highlight the correspondence between the two Petri Nets and the changes stemming from the recommendations, the Petri Nets are formalized in a table (to the left, the procedure envisaged by the software and at the right, the procedure integrating recommendation).

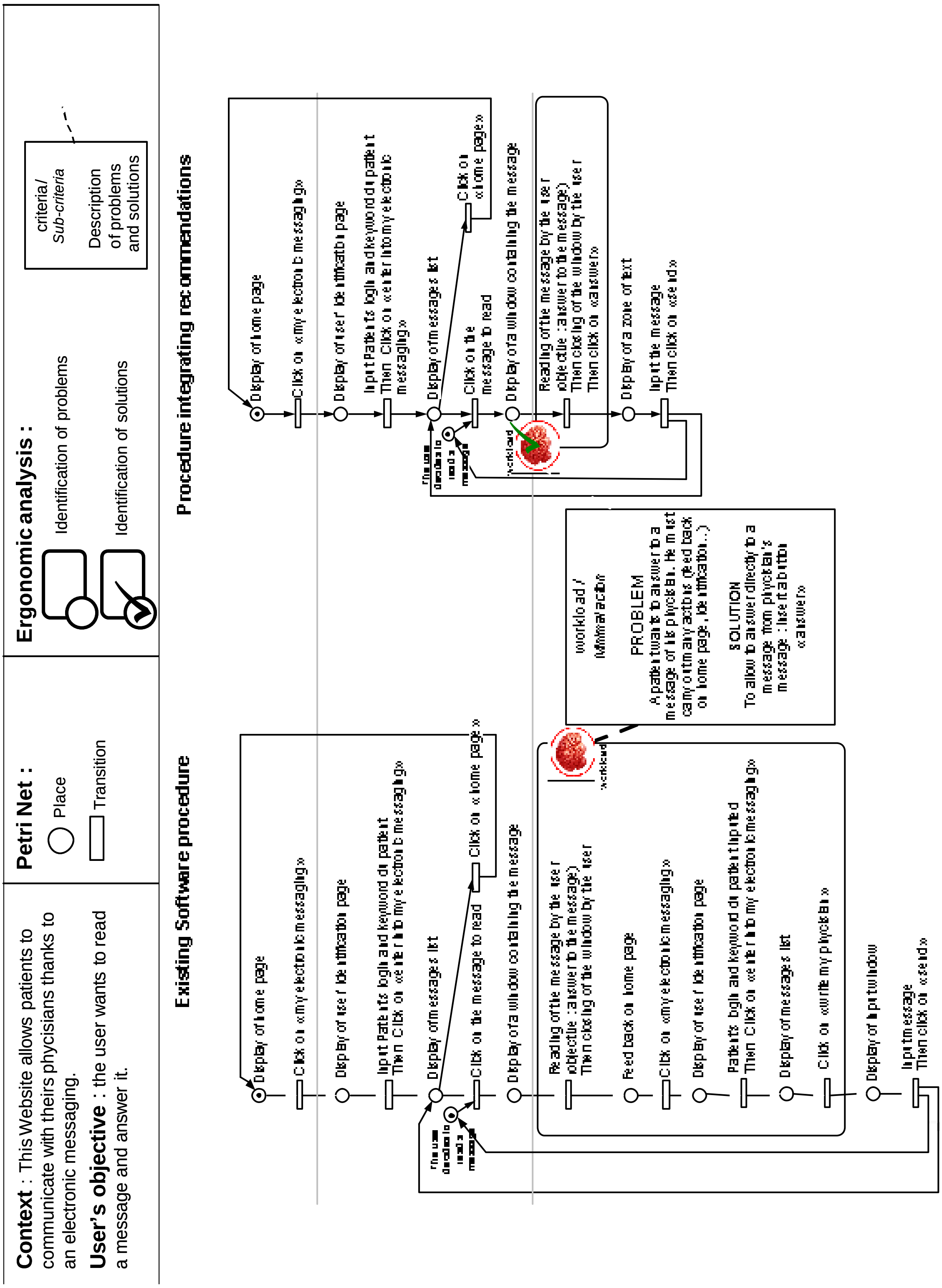

Figure 2. ErgoPNets modeling: assistance for modeling ergonomic problems and extracting solution recommendations.

Each line of this table corresponds to different parts of the procedures. Comparing the Petri Nets, particularly the last line, shows that the right side is shorter than the left side. (5) *Identification of the recommendations in the procedure*: To identify the recommendation in the Petri Net at the right, we have circled the modifications. The "workload" icon used in the step 3 (Petri Net at the left) is employed again, and a validation sign (✓) is added to the icon. Finally, a text explaining the proposed solution is added in the text zone corresponding to the problem.

5 Conclusion and perspectives

ErgoPNets is the result of associating Petri Nets and ergonomic criteria. This method allows a detailed though less ambiguous modelling of ergonomics problems and recommendations, through the use of procedures. Indeed, it situates the detected problems precisely in the first Petri Net and proposes a solution proposed by human factors specialists in the second Petri Net. Moreover, these specialists can verify their recommendations by constructing mock-ups since the Petri Nets allow the logical running of the solution to be described and highlights the possible inconsistencies in the procedures. In addition, ErgoPNets are entirely appropriate when the problem detected requires a detailed description.

This method has already been validated during the re-engineering of a medical IT application. However, the underlying research continues. The tool could be improved to better help users and to document projects. In addition, we envisage an activity analysis of human factor specialists and computer specialists, using ErgoPNets to study how the method can be integrated into their activities and to identify the method's strengths and weaknesses. Finally, we hope to integrate this method in well-known general software engineering methods, such as UML, or at least to provide a way to associate ErgoPNets with such general methods.

Acknowledgement

This research was supported financially by the FEDER, the *Nord/Pas-de-Calais* Region (the TAC MIAOU and EUCUE Projects), and the French Ministry of Research (the RNTS and Presc'Info projects).

6 REFERENCES

1. Abed M, Bernard JM, Angué JC (1992) Method for comparing task model and activity model. Proceedings 11th European annual conference Human Decision Making and Manual Control. Valenciennes, France.
2. Abed M (2001) Méthodes et modèles formels et semi-formels pour la conception et l'évaluation des systèmes homme-machine. Mémoire d'HDR, University of Valenciennes and Hainaut-Cambrésis, Mai.
3. Bastien JMC, Scapin LD (1993) Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. Rapport technique INRIA, n°156.
4. Bastien JMC, Scapin LD (1994) Evaluating a user interface with ergonomic criteria. Rapport de recherche INRIA n°2326.

5. Bernonville S, Kolski C, Beuscart-Zéphir MC (2005) Contribution and limits of UML models for task modelling in a complex organizational context: case studyin the healthcare domain. In K.S. Soliman (Ed). Internet and Information Technology in Modern Organizations: Challenges & Answers, Proceeding of The 5th International Business Information Management Association Conference (December 13-15, 2005, Cairo, Egypt), IBIMA, pp 119-127.

6. Beuscart-Zephir MC, Pelayo S, Anceaux F, Meaux JJ, Degroisse M, Degoulet P (2005) Impact of CPOE on doctor-nurse cooperation for the medication ordering and administration process. International Journal of Medical Informatics 74:629-641.

7. Card SK, Moran TP, Newell A (1983) The psychology of human-computer interaction. Hillsdale, NJ : Erlbaum.

8. Da Silva PP, Paton NW (2000) UMLi: the Unified Modelling Language for interactive applications. Proceedings 3rd International Conference on the Unified Modeling Language UML'2000, LNCS, vol. 1938, Springer-Verlag, Berlin, pp. 117-132.

9. Ezzedine H, Kolski C (2005) 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.

10. Mayhew DJ (1999) The usability engineering lifecycle. Morgan Kaufmann Publishers.

11. Nayak NP, Mrazek D, Smith DR (1995) Analyzing and communicating usability data: Now that you have the data what do you do? ACM SIGCHI Bulletin, 27:22-30.

12. Nielsen J, Molich R (1990) Heuristic evaluation of user interfaces. Proceedings of ACM Conf. CHI'90, ACM Press, Seattle, pp 249-256.

13. Pelayo S, Leroy N, Guerlinger S, Degroisse M, Meaux JJ, Beuscart-Zephir MC (2004) Méthodes ergonomiques pour soutenir la ré-ingénierie des applications logicielles en santé : l'exemple des fonctionnalités de prescriptions thérapeutiques. Proceedings of ErgoIA, Biarritz, France, Novembre.

14. Palanque P, Bastide R (1997) Synergistic modelling of tasks, system and users using formal specification techniques. Interacting With Computers, 9:129-153.

15. Palanque P, Farenc C, Bastide R (1999) Embedding Ergonomic Rules As Generic Requirements in a formal Development Process of Interactive Software. In Proceedings Interact'99, Sasse A, Jonhson C (Eds). IOS Press, pp 408-416.

16. Schael T. (1997) Théorie et pratique du workflow. Des processus métier renouvelés. Springer Edition, New York.

17. Tabary D, Abed M (2002) A software environment task object oriented design (ETOOD). Journal of Systems and Software, 60:129-140.