Published in: **International Journal of Medical Informatics** (IJMI), Vol. 79, Issue 4, 2010, pp. 35-42.
(final version available in ScienceDirect)

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

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

[1] LAMIH-UMR CNRS 8530, University of Valenciennes and Hainaut-Cambrésis, Le Mont Houy, F-50313 Valenciennes cedex 9, France
[2] EVALAB, EA 2694, Centre Hospitalier Régional Universitaire de Lille, Faculté de médecine, 1 place de Verdun, 59045 Lille, France

Abstract
The integration of Software Engineering (SE) and Human-Computer Interaction (HCI) methods and models is an interesting means for modelling an organization's activities, with software applications being part of these activities. It is particularly relevant for organizations concerned by current or future Computerized Physician Order Entry systems for medications. Integrating these SE and HCI methods and models allows case studies to be seen from the technical, organizational and ergonomic perspectives, and may also make it easier to compare current and future work situations. These techniques can be exploited to translate the data provided by Human Factors specialists by creating efficient communication supports that can be easily understood by all project partners, particularly computer scientists, which will facilitate software re-engineering or design. In this paper, the basic principles behind such communication supports are described and illustrated by a real case study.

Keywords
Human-Computer Interaction (HCI), HCI modelling, design or re-engineering project, human factors, Computerized physician order entry (CPOE)

Introduction
The hospital activities of diagnosis and ordering/dispensing medications display all of the characteristics of a very complex sociotechnic system [1]. Indeed, physicians and nurses work in a large problem space, where a lot of different people work in a dynamic situation that has a high potential hazard level. The softwares that support this type of activity, called CPOE (Computerized Physician Order Entry) softwares, must take of all the constraints engendered by this complexity into account. CPOE softwares are very often seen as a means of reducing medication errors, specifically by helping to avoid recopying errors or by carrying out automatic checking of drug interactions [2]. However, implementing and using such tools are very difficult. In fact, if the tool is not perfectly adapted to the users' activities, they can and do refuse to use the tool [3] or make new types of medical errors [4] when using it.

Taking into account the human factors more effectively in the design and implementation phases of CPOEs should make it possible to improve the softwares' safety and make CPOEs more acceptable to users. In order to guarantee that ergonomic factors will be taken into account, human factors specialists must be integrated in a user-centered design phase [5]. However, communication between the human factors specialists and the technically-oriented project team is often problematic, which conditions the way in which human factors are taken into account and integrated in the system. Thus, for interactive systems design or re-engineering projects, it is important that the human factors specialists and the software developers/design team work in close cooperation. Because the different project partners all have their own domain-specific vocabulary and methods [6], dialogue and comprehension is frequently hampered, thus making cooperation difficult. In addition, the existing methods and models are often inadequate for dealing with complex socio-technical systems.

One possible solution to this communication problem is for the various project actors to create common work supports, using Software Engineering (SE) and Human-Computer Interaction (HCI) methods and models (figure 1). To try to implement this solution, we studied the methods and models that seemed most interesting to us, adapted them when necessary and combined them to produce other methods.

In this paper, we first demonstrate the advantages of using the SE/HCI methods and models when designing or re-engineering interactive systems that support user activities in complex organizations. We then describe how we created the common work support and explain its potential for use in such projects. Finally, we present a detailed example of a common work support created using the ErgoPNets method, illustrated by a specific problem related to a CPOE.

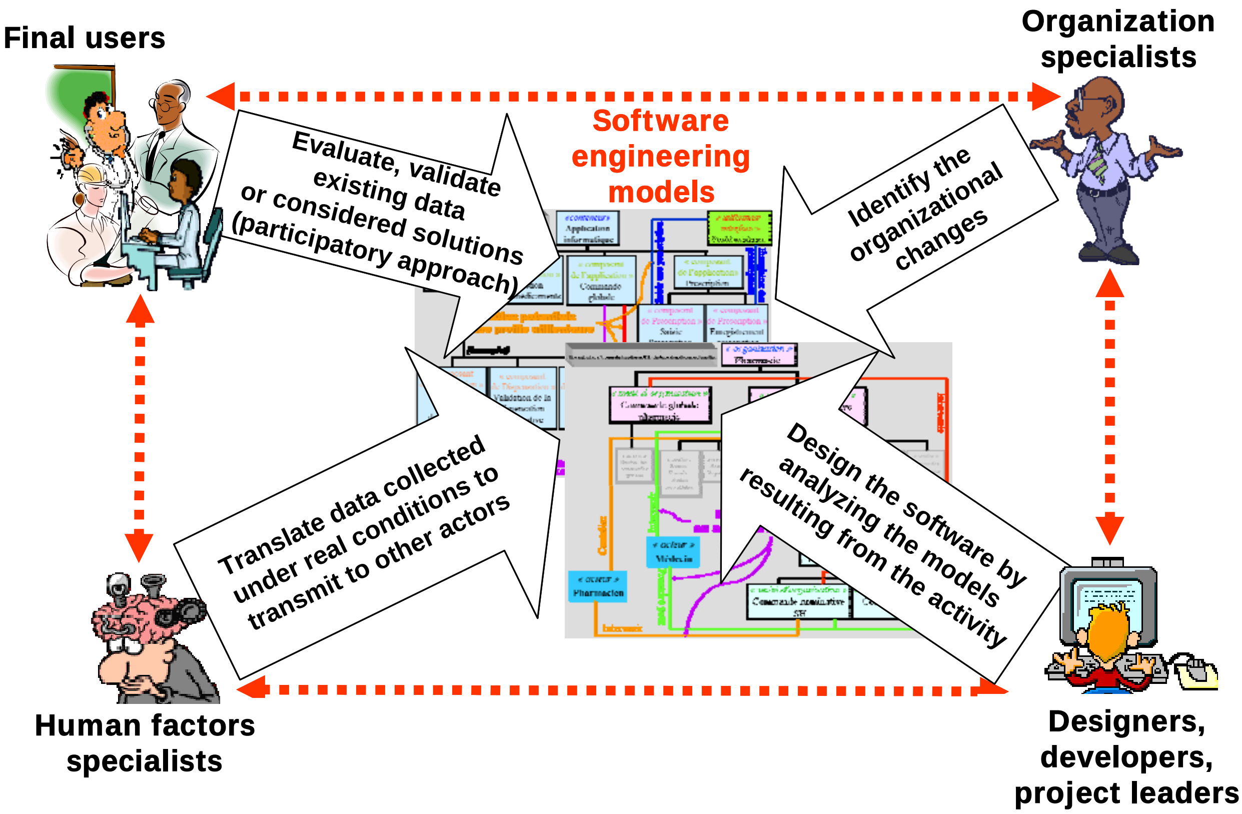

Figure 1: contributions of modelling

The Need for SE & HCI methods and models

In this section, we attempt to demonstrate the potential of SE & HCI methods and models both for analyzing and modelling business processes in which CPOE tools are envisaged or evaluated and for making communication easier in such projects.

Indeed many models have been designed specifically so that they can be understood by the different people participating in the project, including both specialists in computer science and specialists in other non-computer-related fields. This universality is achieved by basing the models on a restricted set of concepts. For instance, a descriptive activity model based on only four concepts (e.g., events, synchronisation, operations & results) can be easily understood by those involved in therapeutic prescription, providing, of course, that they receive a minimum of training beforehand. In fact, for many models, the necessary training need only last a few minutes.

We think than these business processes can be considered from the angle of information systems analysis and design, with an emphasis on the organization. To this end, Software Engineering proposes systemic methods (e.g., MERISE and its variants, OSSAD and its variants) that include a set of models centered both on the information exchanged or exploited in these organizations and on the manual, interactive and automatic treatment of the data. In addition, because they focus on very specific concepts, other methods that not are centered on the organization can provide interesting models from a variety of points of view:

(1) object-oriented approaches using UML (www.omg.org) offer nine basic models and extensions that are widely exploited companies, which can be gradually associated with methodological processes that explain how these models should be implemented in software analysis and design projects,
(2) cartesian methods, of which SADT (Structured Analysis and Design Technique) is most representative, provide a way to model the activities of the various project actors, be they human, hardware or software.

In addition, Petri nets can also be used for modeling dynamic software applications, as will be explained further later in this text. Moreover, elements from several methods (e.g., models, approaches) can be combined to create a composite method that meets specific needs, in which case Software Engineers speak of integrated methods.

In the field of Human-Computer Interaction, specific methods, or extensions of the methods mentioned above, have been proposed to provide solution to problems encountered when designing and evaluating interactive systems, specifically to analyze and model human tasks and activities and to describe user needs [7] [8]. To move towards more user-centered approaches to design and evaluate CPOE tools that must be naturally and effectively integrated into business processes, such HCI methods must be taken into account.

The approach adopted

Projects involving the design or re-engineering of interactive systems call upon the skills of ergonomists who use a variety of data-collecting methods, such as on-site observations, interviews, and ergonomic inspections (For more information, see reference [9]). Once the data has been collected, the elements that are relevant to the project must be summarized and transmitted to the various partners. The existing techniques for modelling complex organizations often have limitations that can be remedied by applying certain aspects of Software Engineering and Human-Computer Interaction methods and models. To this end, we studied a set of SE & HCI methods and models and retained those that seemed most suitable, each one of them taking specific elements into account.

Table 1 provides examples of the SE & HCI methods and models used to create the supports used in interactive system design or re-engineering projects. The five columns of this table represent the various steps of our procedure: (1) the starting data; (2) the elements that must/should be modelled (e.g., actor, activity, ergonomic problems); (3) the selected method or model (e.g., UML activity diagram, ErgoPNets method); (4) the possible uses of the supports created; and (5) the phase of the human factors engineering process (i.e., analysis of the demand, analysis of the work situation, cooperative design, iterative evaluation and monitoring; see [6] for more information about this process). We choose the SE/HCI methods and models listed in column 3 to correspond to particular modelling needs. For example, an UML activity diagram allows the following elements to be modelled: objects, swimlanes, actions, flows, decisions disconnections, synchronization, initial states, & final states. Such a systematic census then enabled us to identify the adapted model (or group of models), by studying the elements to be modelled (contained in the second column of table 1). For example, for the first example of table 1, the selected model is the UML activity diagram because elements to be modelled can correspond to specific elements taken into account by this model (such as actors and swimlanes, activity and actions, documents used and UML objects).

Starting data	Elements to be modelled	Selected SE and HCI methods and models	Possible uses	Phases of the human factors engineering framework concerned
Data not structured in the form of: - a coding scheme from observations, interviews (textual data, audio files, video files) - Tables / inventories (roles, actors, tasks, procedures, contexts, data supports used, location…) - hierarchical diagrams representing tasks, sub-tasks and actions - Statistical data (time…) - a coding scheme from documents used during the activity (data categorization and quantification)	- Actors - Activity - Action and decision-making sequences - Documents used - Situations (non-computerized vs. computerized)	**UML activity diagram** (with some adaptations) **Modelling elements taken into account:** - Swimlanes - Actions - Flows - Decisions - UML Objects - Forks - Synchronisation - Initial states - Final states	- Comparison of work situations with and without the software tool - Description of the changes in the organization - Contribution to the re-design of the work situation	Work situation analysis
- List of ergonomic problems categorized by ergonomic criteria (textual description of problems and consequences, screen shots, degree of gravity) - Recommendations (textual description of recommendations, mock-ups) - Test results (lists of problems, problem quantification, video files)	-Ergonomic problems linked to the procedures and envisioned in the existing software - Recommendations associated with the detected ergonomic problems	**ErgoPNets Method** (elaborated by Bernonville et al. [11]) **Modelling elements taken into account:** - System states - Transitions (conditions or events) - Graphic descriptions - Procedures - Ergonomic criteria - Petri net properties (to verify possible inconsistencies)	- Detection of procedural inconsistencies - Improved comprehension and interpretation of problems and recommendations - Project documentation	- Iterative evaluation - Work situation analysis

Table 1*: Extract of the synthesis illustrating our procedure, based on SE and HCI methods and models*

Creating a common work support using the ErgoPNets method

When designing or evaluating an existing software, ergonomists must detect various types of problems, ranging from simple problems with the graphic representation (e.g., the color of a button) to more complex problems involving the procedure that the software is supposed to facilitate. To describe these problems, as well as their ergonomic solutions, to the designers and developers, ergonomists tend to use textual descriptions, accompanied if necessary by proposal mock-ups. However, textual description can lead to problems of comprehension and interpretation, especially when dealing with the complex problems involving procedures. In fact, textual explanations can be ambiguous, which leads to erroneous interpretations, which in turn could lead to software modifications that do not correspond to the intended recommendations. ErgoPNets allow ergonomic problems and recommendations to be described unambiguously, facilitating both comprehension and interpretation.

The following section presents the ErgoPNets method, from the choice of the method to the description of an ergonomic problem detected on an existing CPOE software to the ergonomists' recommendations for solving this problem.

The ErgoPNets method

Choice of the method

The starting data given in table 1 include a list of ergonomic problems categorized by ergonomic criteria (e.g., textual description of problems and consequences, screen shots, degrees of importance) and recommendations (e.g., textual description of recommendations, mock-ups). The ergonomist must describe an ergonomic problem involving a software procedure and wants to enhance the mock-ups of results of the proposed recommendations. The method selected is the ErgoPNets method, which is characterized by the use of Petri nets (PN) and explanations about problems and recommendations located directly on the described procedures.

The PN takes the following elements into account: system state (e.g., home page displayed), condition or event (e.g., click on the "validate" button), graphic description of the system states (represented as circles) and conditions or events (represented as rectangles), procedure (sequence of states and conditions or events), and proprieties (e.g., verifying the compliance with PN construction rules). Ergonomic criteria, represented by icons (one icon for each criterion), allow the category of the ergonomic problem to be visualized on the PN. The ergonomic criteria used here are taken from the work of Bastien and Scapin [11], though other criteria could have been used.

Principles of the method

To realize a common work support, five steps are necessary: (1) define the user objective, (2) describe the existing software procedure that currently is used to fufill the objective using an adapted PN formalism, (3) identify and explain the detected ergonomic problems using icons representing Bastien and Scapin's ergonomic criteria, (4) describe the procedure for integrating the ergonomists’ recommendations, as illustrated by the mock-ups, again using an adapted PN formalism**,** (5) identify and explain the recommendations using icons representing Bastien and Scapin's ergonomic criteria. (For more detailed information about the ErgoPNets method, the interested reader should refer to reference [12].)

Use of the method

The ErgoPNets method allows:

(1) the enhancement of ergonomist mock-ups with representations of system dynamics, which in turn allows the ergonomists to validate their recommendations during the mock-up construction process;
(2) the use of procedures and the location of problems and recommendations on the described procedures, making it possible to reduce the ambiguity that makes

Published in: **International Journal of Medical Informatics** (IJMI), Vol. 79, Issue 4, 2010, pp. 35-42.
(final version available in ScienceDirect)

designer/developer comprehension and interpretation difficult, thus facilitating concrete solutions to the identified problems; and

(3) the documentation of project which allows to keep written trace and to reuse documents in other projects (e.g. the description of a procedure illustrated by a mock-up).

Applying the ErgoPNets method to a CPOE software

This section described our application of the ErgoPNets method to a CPOE software. Figure 2 shows an annotated screen shot of the input data page for a drug order filled out by the attending physician. We can see the different steps followed by the physician:

(1) the desired drug is searched in the medical database using the drug database search. (In figure 2, the search zone in retracted because the search was already carried out.)
(2) the search results are displayed, with a checked box indicating the drug chosen,
(3) the drug is ordered using the data entry form,
(4) the order is recorded using the "save" button.
(5) Each physician order in a same session is automatically added to and saved in a virtual basket, which can be displayed on the screen if the physician clicks on the "order-in-progress basket" button.

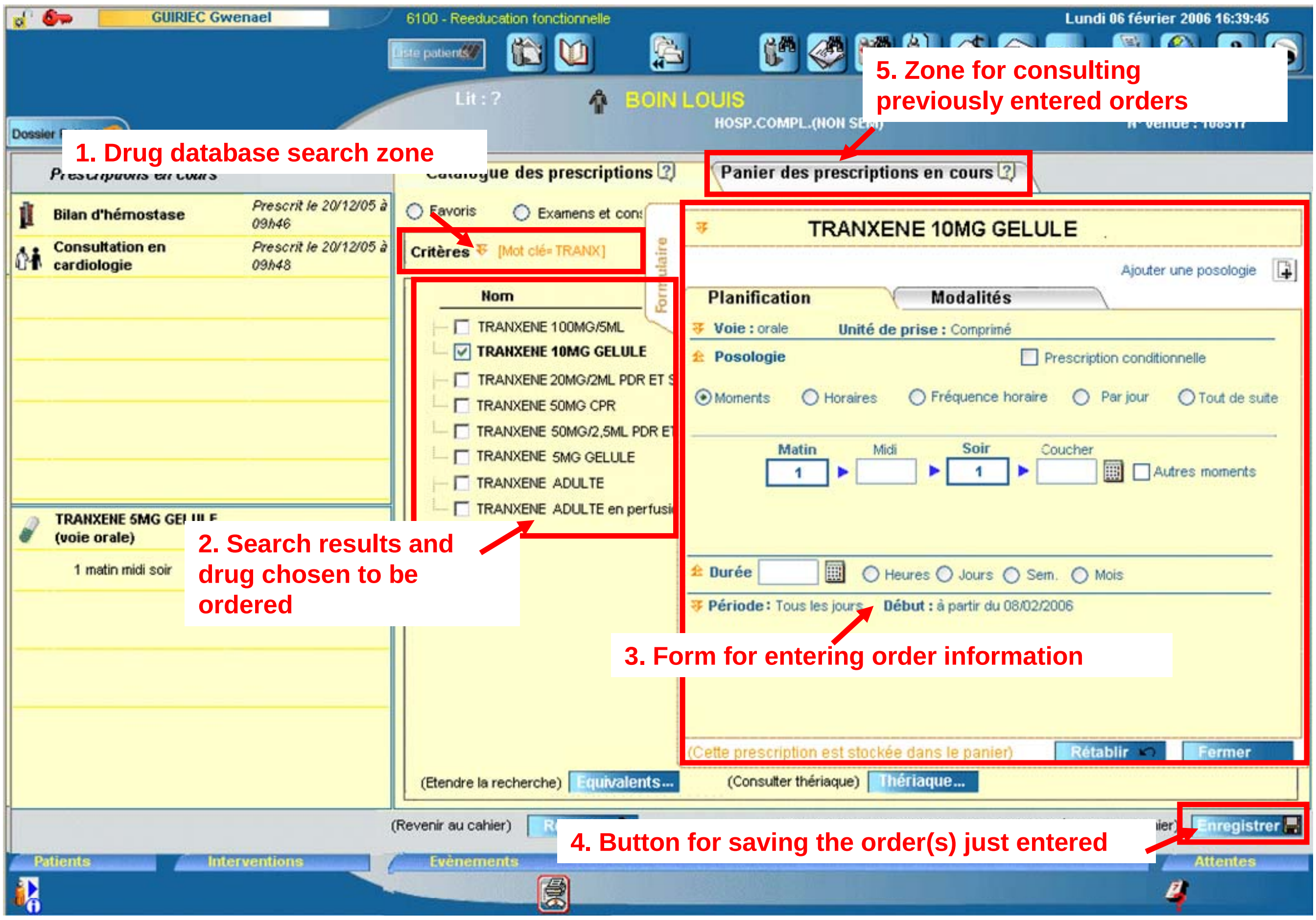

Figure 2. *Screen shot of the CPOE* (page for entering order data*)*

Published in: **International Journal of Medical Informatics** (IJMI), Vol. 79, Issue 4, 2010, pp. 35-42.
(final version available in ScienceDirect)

During the ergonomic evaluation of the software, the ergonomists detected the following problem: When physicians save their orders using the "save" button, the system immediately displays the updated patient treatment. It does not encourage the physicians to verify the information that they enter; they must remember to click on the "order-in-progress basket" button, which they might not do. To remedy this problem, the ergonomists recommended displaying the virtual basket with the orders just entered when the physician clicks on the "save" button. Thus, the physicians can verify the information entered one last time and make any necessary corrections. After verification, the physician can confirm the order by clicking on the “save the basket” button.

Figure 3 shows the ErgoPNets method's depiction of the problem and the recommendation described above. To the left of the figure, the existing software procedure is shown and the place where the problem occurs is indicated. To the right of the figure, the procedure integrating the ergonomists recommendation is shown, as well as the level at which the recommendation enters into the procedure. In the middle of the figure, we added a text zone that presents the problem type according to Bastien and Scapin's ergonomic criteria as well as a textual description of the problem and the recommended solution. The support obtained allows the designer/developer to better understand the problem and provides a clear and precise solution.

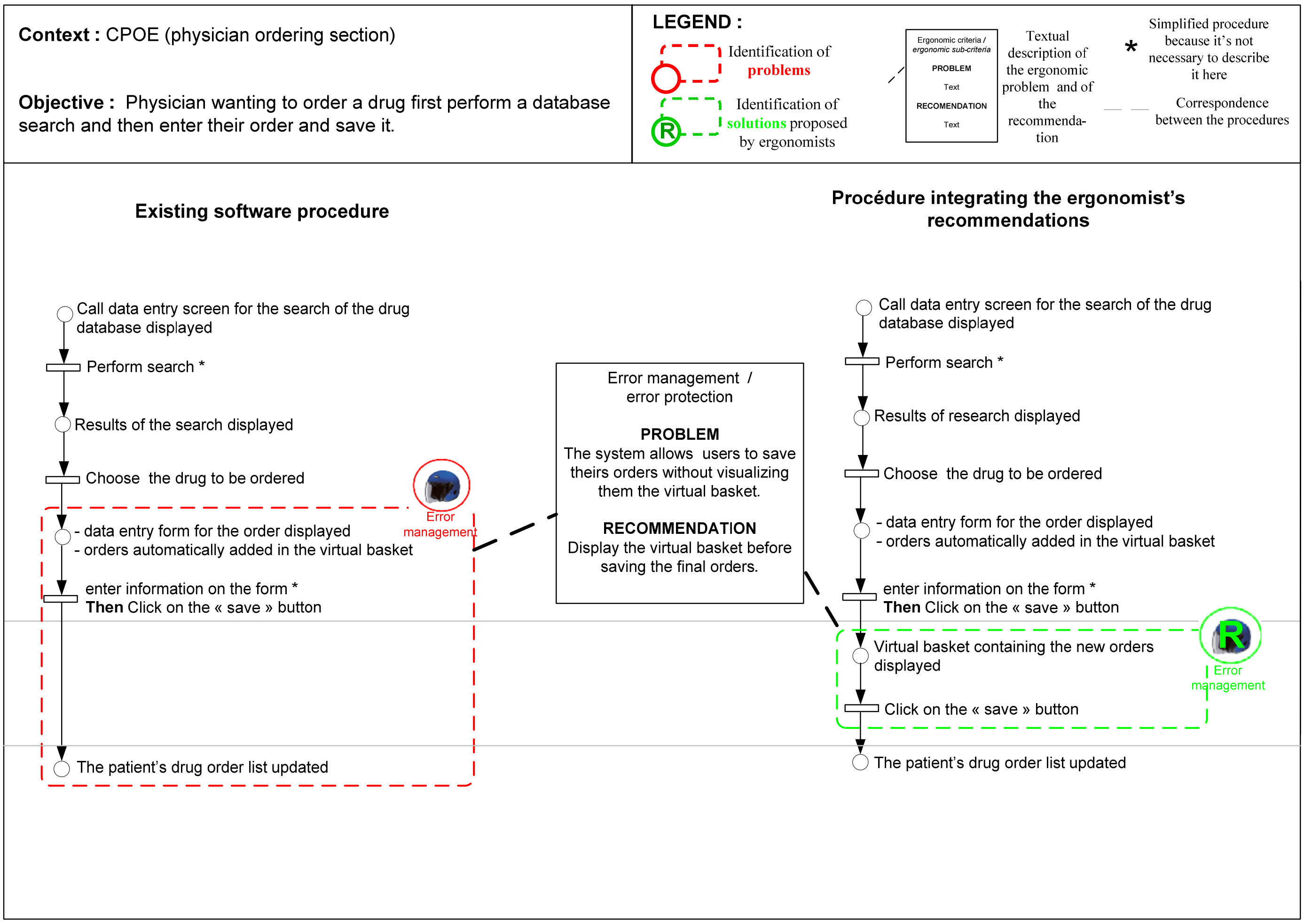

Figure 3: Example of the support created with the ErgoPNets method

Conclusion

It is essential to facilitate the communication in projects seeking to design or re-engineering interactive tools, like CPOE. To this end, we studied the potential contribution of a set of Software Engineering and Human-Computer Interaction methods and models. Based on this survey, we adapted some of them and rejected others, instead suggesting new models. Both the new and adapted models are now being applied to real projects. In the future, we hope to better articulate the correspondence between these models and to implement a software workbench to help choose the most appropriate model according to the characteristics of a project.

Acknowledgements

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

References:

1. Vicente, K.J., Cognitive Work Analysis, Mahwah, NJ: Lawrence Erlbaum Associates, 1999.
2. Kaushal R., Bates D.W., Information technology and medication safety: what is the benefit?, Qual Saf Health Care 2002;11(3):261-265.
3. California Healthcare foundation, Cedars-Sinai suspends CPOE use, January 2003, available at http://www.ihealthbeat.org/index.cfm?action=dspItem&itemid=117155.
4. Ammenwerth E., Talmon J., Ash J.S., Bates D.W., Beuscart-Zephir M.C., Duhamel A., Elkin P.L., Gardner R.M., Geissbuhler A., Impact of CPOE on mortality rates--contradictory findings, important messages, Methods Inf Med 2006; 45(6):586-93.
5. Beuscart-Zéphir M. C., Elkin P., Pelayo S., Beuscart R., The Human Factors Engineering approach to biomedical informatics projects : state of the art, results, benefits and challenges, In press, IMIA Yearbook of medical informatics 2007, special issue
6. Nayak N.P., Mrazek D., Smith D.R., Analyzing and communicating usability data: Now that you have the data what do you do?, ACM SIGCHI Bulletin 1995; 27, 22-30.
7. Diaper D., Stanton N., Handbook of Task Analysis for Human-Computer Interaction, Lawrence Erlbaum Associates, London, 2003.
8. Jacko J.A, Sears A., The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications (human factors and ergonomics), Lawrence Erlbaum Associates, London, 2002.
9. Mayhew D. J., The usability engineering lifecycle, Morgan Kaufmann Publishers, 1999.
10. Bastien J. M. C., Scapin D. L., Evaluating a User Interface with Ergonomic Criteria, International Journal of Human-Computer Interaction, 1995; 7 (2): 105-121.
11. Bernonville S., Leroy N., Kolski C., Beuscart-Zéphir M. C., Explicit combination between Petri Nets and ergonomic criteria: basic principles of the ErgoPNets method, Proceedings to the 25th European Annual Conference on Human Decision-Making and Manual Control (EAM) 2006, Valenciennes, France.