Formalization of Ergonomic Knowledge For Designing Context-Aware Human-Computer Interfaces

Dorra Zaibi
Cristal Laboratory
National School of Computer Sciences
University of Manouba
Manouba, Tunisia
e-mail: zaibi.dorra@gmail.com

Meriem Riahi
Higher National School of engineering of Tunis
University of Tunis
Montfleury, Tunis, Tunisia
e-mail: meriem.riahi2013@gmail.com

Faouzi Moussa
Cristal Laboratory
National School of Computer Sciences
University of Manouba
Manouba, Tunisia
e-mail: faouzimoussa@gmail.com

***Abstract*—During the last three decades, different methodologies and tools were developed for the design and the evaluation of Human-Computer Interaction. However, ergonomic knowledge is often informal and lacks - for the utmost part any structure to be directly exploited by designers. Ergonomists still find it hard to use the ergonomic requirements in practice. This paper presents an ergonomic approach for designing Human-Computer Interfaces considering context information. We are particularly interested in formalizing the ergonomic knowledge. To validate our approach, we propose the application of a Computerized Maintenance Management System, dedicated to the management of intervention requests and dashboard monitoring.**

Keywords- Ergonomic knowledge; expert system; Human computer Interaction (HCI), Model Driven Architecture (MDA); context-aware design.

I. INTRODUCTION

Nowadays, interactive systems are continuously evolving. They are pervasive in all areas and fields [1]. Additional functionalities further increase the complexity of these systems making them less suitable to use [2]. Thus, the Human-Computer Interaction (HCI) community is facing a growing challenge to obtain high quality User Interfaces (UI) [3][4]. To this aim, several novel methodologies and techniques were presented and explored during the last three decades [5][6][7]. However, as stated in Scapin, Reynard and Pollier [8], ergonomic knowledge is usually rather informal and lacks - for the utmost part - structuration to be directly applied by designers. It still remains a difficult task to ergonomists for the ergonomic recommendation practice.

Many directions were explored. They differ by means of interactive system life cycle phase (specification, design, development, testing, validating, etc.). Different quality factors were used, over which the evaluation process can be established. The most frequently used one for interactive system evaluation are essentially the utility and the usability [9][10]. As reported by Charfi et al. [11], “the utility determines if the system allows the user to perform his task and if he is able to achieve what is necessary to meet his expectations from the system. It corresponds to the functional capabilities, system performance, and the technical assistance quality given to the user by the system”. According to ISO 9241-18, the term usability has been defined 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” [12]. The utility and usability are important issues to validate ergonomic UI and assure optimal functioning of Human-Computer systems.

Model-Driven Architecture (MDA) [13] has been revealed to be an appropriate approach for the design and the development of software system and their user interfaces. Following this trend, the HCI community has highlighted the benefits of its technology and used MDA approaches.

In this paper, we propose an ergonomic approach for designing context-aware UI based on MDA. We are interested in ergonomic evaluation process since early design stages. This work aims to formalize ergonomic knowledge for the specification of context-aware Human-Computer interfaces.

The remainder of the paper is organized in the following way: Section II surveys the related work. Section III introduces our contribution for ergonomic knowledge formalization and integration in the design process. Section IV presents an overview of our current implementation prototype as well as a case study. Finally, we conclude and discuss our research perspectives in Section V.

II. RELATED WORK

Ergonomic approaches motivated many publications.

Tarby et al. [14] based their approach on traces to conduct design phase evaluation, which include Aspect Oriented Programming (with aspects that are devoted to evaluation).

Trabelsi et al. [15] present the evaluation of agent-based interactive systems. They propose an evaluation approach based on three complementary techniques: assistance evaluation tool, questionnaire, and verbalization.

User Interface evaluation framework (RITA) [16] is an evaluation tool. It is composed by software applications structured in a modular architecture way. This framework uses three different evaluation techniques: the electronic informer, the ergonomic quality inspection and the questionnaire. It has a modular architecture that can be configured to evaluate different kinds of user interfaces.

 ISBN: 978-1-61208-498-5

Destine [17][18] is a tool, which allows the UI evaluation using a language of ergonomic rules definition. It permits to integrate or change intended rules for the evaluation without altering the source code of the evaluation tool.

EISEval [19] is an environment dedicated to the evaluation of interactive systems having agent-based architecture. The authors presented an electronic informer using Petri Nets.

These approaches vary in several ways. Their applications differ with the system lifecycle. Some tools and approaches are proposed to be used during design phases [14][16]. Others are designed to evaluate UI during the last phase of the development life-cycle [15][17][18][19], without considering the design and implementation process of the UI. In fact, we noticed that the UI evaluation phase is skipped in most cases of the development. It may be neglected for different reasons such as time and budget constraints. In addition, the evaluation can be expensive when applying it to the final stage of the lifecycle of a system. According to Nielsen [10], it is a hundred times cheaper to evaluate the UI during the early stages than in the final ones. Thus, it is more appropriate to consider the UI evaluation early in the development cycle. Some existing techniques propose to incorporate ergonomic guidelines in UI design process. Unfortunately, they do not take into consideration context changes. Thus, interactive systems face mostly the challenge of their ergonomic inconsistencies due to context requirements.

The novelty of this work is how to consider the ergonomic guidelines properly with eventually possible changes in the context of interactive systems for obtaining successful UI. We present in the next section our methodology for designing context-aware UI.

III. Human-Comuter Interface Design Methodology : Overview

Since our focus is to provide ergonomic knowledge, we include some ideas to consider ergonomic specifications since early design stage. In this way, it is necessary to first conduct a formalization of ergonomic knowledge, then to provide the appropriate ergonomic guidelines considering context information, and finally to integrate these guidelines in context-aware specifications.

Our approach is summarized in Fig. 1. The system is developed by performing two main steps. We consider that we are given regularly context information. In this sense, a specific system is held, at runtime, updating continuously context changes. During the design phase of the system, we consider this information when designing the UI. The main steps of our approach are explained in the sequel.

A. *Expert System for Reasoning Ergonomic Knowledge*

Ergonomic knowledge is generally informal. For this purpose, we aim to formalize ergonomic knowledge for integrating them in the context-aware specifications. To be used in such approach, an ergonomic knowledge base is supplied. Our methodology is based on Artificial Intelligence techniques. In fact, our solution can be thought of as an expert system. As Fig. 1 shows, we have followed three sub-stages:

1) Structuring of ergonomic knowledge: The ergonomic knowledge that we used is mainly derived from [11][12][20][21].

It addresses more particularly the ergonomist to provide specific structural definition. This organization is important to ease the management of rules database. In this phase, we use the structuration of User Interface Markup Language (UIML) [22], since it allows the description of user interfaces in an abstract way. UIML, as defined in [22], separates the elements of the interface. It contains two main components: *interface* and *peers*. The interface component represents the description of the user interface according to four parts: *structure*, which describes the organization and the hierarchies of all the parts constituting the UI; *content*, which represents the application's information to be shown; *behavior*, which defines the behavior of the application during the interaction with the user, and *style,* which describes all the particular properties for each element of the UI. The peers' component contains two parts: *presentation*, which links the generic elements of the UI to a particular platform and *logic*, which defines the calling conventions for external methods in application logic that the UI raises.

Building on the organization of UIML elements, this stage promotes the semantic description of ergonomic knowledge. At the end of this phase, the knowledge base is shared in "knowledge sub-bases" (*KD Sub-Base*), each sub-base is related to a semantics of the ergonomics: *Structure, Style, Contents, Behavior, Logic and Presentation.*

2) Formalization of ergonomic knowledge: After the structuring phase, the ergonomic knowledge is formalized in form of productions rules. The set of production rules is related to particular context information. The production rules' formalism is of the following type:

IF $(C_1 * C_2 * --- * C_i)$ *THEN E.*

The contextual conditions of a rule are a logic arrangement of premises *Ci*. The operator "*" may be the one of the logical operator "AND" or "OR". The ergonomic guidelines *E* represent the set of appropriate conclusions of a rule.

3) Rule-based Inference System: We are based on a Rule-based system, which is designed to resolve problems by reasoning knowledge, represented initially as if-then rules. The rule-based system uses a working memory that initially comprises the input data for a particular run, and an inference engine.

The set of context instance *Ci* is stored in facts. It is exploited by the inference engine. The context information is defined, initially, in a Context Model (detailed in Section B) and is then explored. The rules deduction is accomplished by the inference of a set of rules contained in the knowledge sub-bases. The ergonomic rules are fulfilled by a first order inference engine that is based on forward chaining inference.

Forward chaining inference engine is an automated reasoning system, working from a current state by evaluating

 ISBN: 978-1-61208-498-5

the premises of the inference rules (IF-part), performing pertinent rules, and then asserting new knowledge.

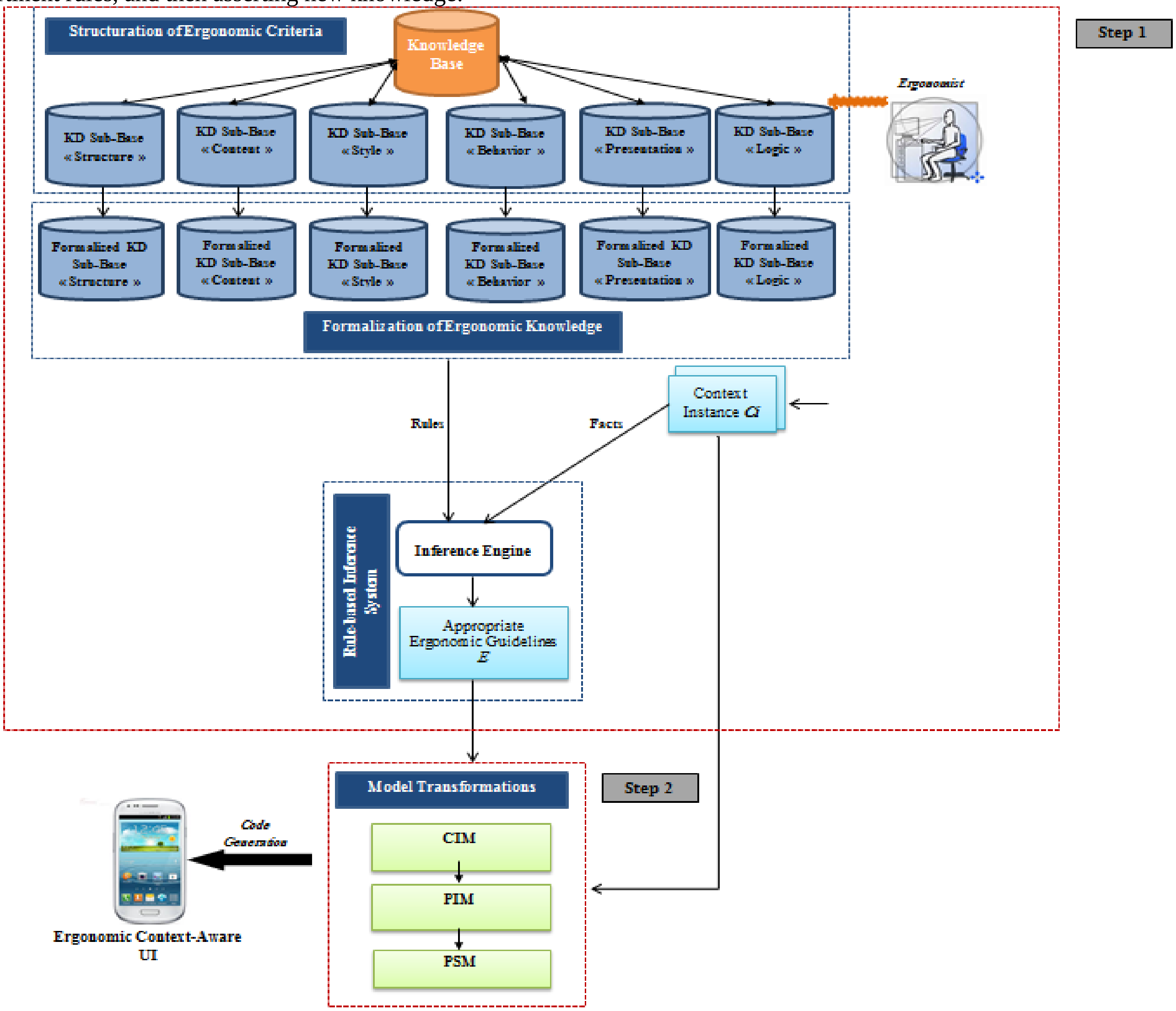

Figure 1. Our Approach overview.

The inference engine draws knowledge based on given facts Ci and production rules. This knowledge presents the set of appropriate ergonomic guidelines for a given context information. The Rule-based system is implemented with DROOLS rule engine [23] using an enhanced implementation of the Rete algorithm [24] for pattern matching, adapted for object oriented languages, and for Java in particular. Rules are stored in the production memory, whereas facts are preserved in the working memory. The Rete algorithm generates a network from rule conditions. Each single rule condition is a node in the Rete network. We use a conflict resolution strategy, if there are multiple selected rules, by managing execution order via the Agenda.

In order to be included on the next step, we have modeled the set of appropriate ergonomic guidelines in an ergonomic model instance (detailed in Section B) using a specific algorithm (i.e., algorithm for converting from a set of appropriate guidelines to a corresponding ergonomic model instance in the Ergonomic Model).

B. Integration of Ergonomic Guidelines in Model-Driven Development

MDA [13] is an approach that allows specifying the system independently of any platform that supports it by transforming the specification into a software system, for a specific platform. A set of guidelines is provided by its approach for the structuring of specifications. It uses three types of models: computation independent model (CIM) that allows describing the context of the system and the requirements without consideration for its structure or treatment; platform independent model (PIM) that allows describing only those parts of a complete specification that can be abstracted without any specific platforms; and, platform specific model (PSM) that associates PIM

 ISBN: 978-1-61208-498-5

specification with specific information for a specific platform.

Following the MDA structure, we distinguish four main models (Fig. 2): The Business Process Model (as CIM), Platform Independent Model 1 (as PIM), Platform Independent Model 2 (as PIM), and Platform Specific Model (as PSM).

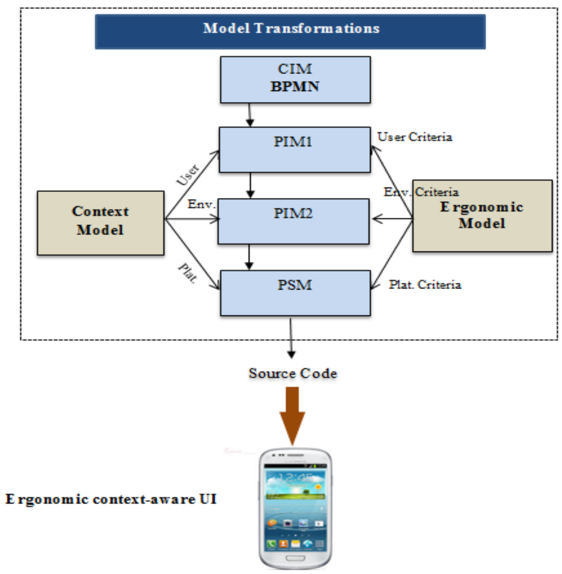

Figure 2. Model-Driven Development .

As described in Fig. 2, this step is composed of six models:

- Business Process Model (BPM): To describe the CIM model, we used Business Process Modeling Notation (BPMN) [25], which is able to define tasks. for the specification of the business goal (interactive tasks, non-interactive tasks and manual tasks) and the information flow among tasks, since we are interested in providing ergonomic guidelines into user interaction. Each BPMN element has been defined in the BPMN model with a specific type (i.e., interaction element type) in order to describe the nature of the interaction with the user, such as: types of input information (e.g., *UIFieldInMultipleChoice*), output information (e.g., *UIFieldInAction*) or grouping information (e.g., *UIUnit*). (Table I);

- Context Model: We have used a defined Context Model all along the MDA model definition and transformations. It is composed of a description about user, environment and platform;

TABLE I. SOME INTERACTION ELEMENTS

BPMN Element	Associated Interaction Element
Pool	UIGroup
Lane	UIUnit
UserTask	UIFieldAction
UserTask	UIFieldInManual

- Ergonomic Model: Fig. 3, below, defines the Ergonomic Model. It is composed of two major parts: *Antecedent*, which is the first half of the hypothetical proposition and contains context information; and *Consequent*, which describes the ergonomic guidelines;

- Platform Independent Model 1 (PIM1): To specify the user interaction by introducing the user requirements and ergonomic guidelines according to the user independently of any platform. It is specified using the User Interface Markup Language UIML [22], specially the *structure* and *style* parts;

- Platform Independent Model 2 (PIM2): To specify the user interaction by taking into account environment constraints and appropriate ergonomic guidelines according to the environment and independently of any specific platform. It is also defined by UIML language [22], which specifies the *style* parts;

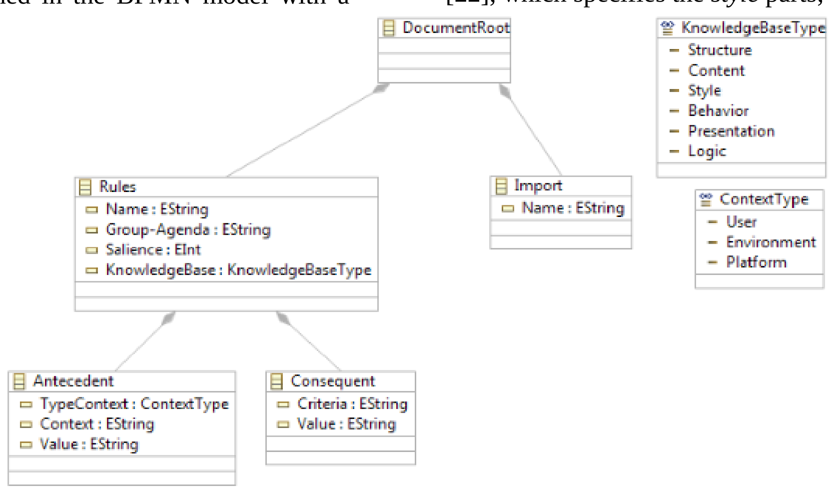

Figure 3. Ergonomic Model

 ISBN: 978-1-61208-498-5

- Platform Specific Model (PSM): To specify the user interaction for a particular platform, integrating ergonomic guidelines related to platform dimensions (e.g., displays information). Moreover, it is described by the use of User Interface Markup Language UIML [22], and allows specifying the *presentation*, *logic* and *style* parts.

By applying Model-To-Model transformations, we generate the first transformation, PIM1 model, based on two UIML parts (*structure* and *style*), which are independent of the platform and integrate the specific information of the user context and its ergonomic guidelines. The second transformation from PIM1 to PIM2 includes environment information and its ergonomic guidelines and generates the *style* part. The third transformation from PIM2 to PSM is generated based on three UIML parts (*presentation*, *logic* and *style*) by including the platform's specific characteristics and its ergonomic guidelines (i.e., according to the platform context). Finally, the last transformation from PSM to the source code is performed, to generate the final UI, supported by existing tools supporting the code generation from UIML (e.g., Acceleo [26]).

IV. Case Study

We have developed our methodology for a simplified case study that implements a Computerized Maintenance Management System (CMMS) application dedicated to the management of intervention requests and dashboard monitoring. In such system, production agent, technician and administrator must be taken into consideration to represent the context of the application. After signing up, a user having technical problem needs to send a request for a new intervention. Therefore, the system shows a detailed form with the required information: *Intervention N°, Sender, Receiver, Desired Date, Intervention Type, Priority, State, Equipment and Observations*. Fig. 4 presents a part of Business Process Model related to this scenario of the process.

Ergonomic knowledge are, initially, stored in a database. We formalized these knowledge considering user, environment and platform dimensions. Since CMMS applications focus on industrial sectors, the user may be localized in a factory. We consider that an industrial environment is typically too noisy. The noise of running machinery is prevalent. In consequence, the system should increase the volume for the notification sound. In the contrary, we want a low sound volume, when a user is having a meeting. Moreover, in order to ensure more visibility for users who have visual problems, the system provides the background with white color theme (i.e., by default, the background is a dark color theme), font size 18 (i.e., the default size is 'medium' or size 12), bold font and activates the zooming options. A beep is emitted when a user has simultaneously a visual problem and is in a meeting. In addition, we consider different platforms with either wide or restricted screens (e.g., in the case of a small screen space, we only display the relevant information, such as Intervention N°, Sender, Receiver, Desired Date, Equipment and Observation).

We consider a sample scenario that describes a user using the CMMS application. As an administrator profile, we chose one that is visually impaired, using the application through a smartphone during a meeting. Using an inference engine (i.e., DROOLS Engine), we inferred a set of appropriate ergonomic guidelines. Fig. 5 presents the appropriate ergonomic guidelines resulted.

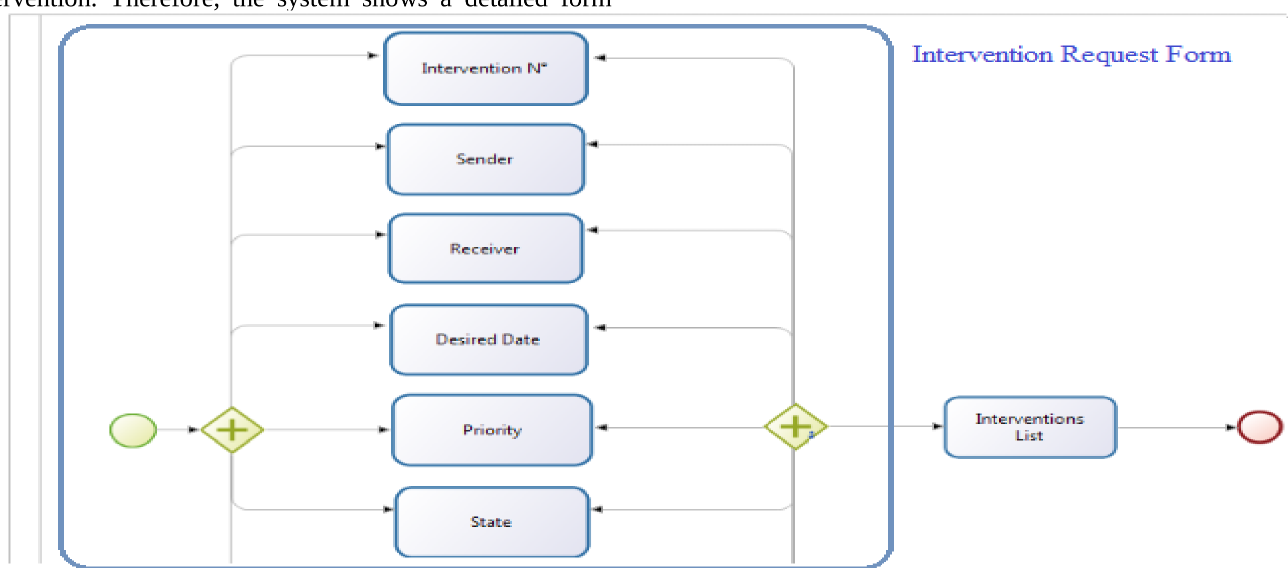

Figure 4. A Part of Business Process Model related to the Intervention request

```
background= WhiteColorTheme
font_size= 18
font_weight= Bold
Zooming_option= true
Volume_control= beep
```

Figure 5. Example of the generated ergonomic guidelines.

After generating appropriate ergonomic guidelines with the consideration of specified context information, the designer should integrate these guidelines within the transformation models. We have implemented model transformation with ATLAS Transformation Language (ATL) [27]. Fig. 6 describes the structure part of PIM1, generated by the ATL transformation rule. Fig. 7 shows the style part of PIM2, generated for both user and environment context. The style contains a list of properties and values. The properties are associated with parts within the UIML document through part-name elements. For example, for the “Intervention_Request_Form” form defined in the <part> element (G:Area class part in Fig. 6), we have defined the background of the form with “WhiteColorTheme” attribute in the property element of Style part. We also specified the “Intervention_N°”, “Sender”, “Receiver” labels, with font size 18 and bold font style.

The generated ergonomic context-aware UI for dashboard monitoring is illustrated in Fig. 8. The UI provides the current user with a rich set of indicators and charts. In Fig. 8, one chart per line is displayed, for enhancing the visibility. Moreover, the zooming option is activated, when selecting the chart for more information.

```
<Structure>

  <part class="G:Area" name="Intervention_Request_Form ">
  ..
   <part class="G:Label" name="Intervention_N°"/>
   <part class="G:TextField" name="Sender"/>
   <part class="G:TextField" name="Receiver"/>
   <part class="G:TextField" name="Desired Date"/>
   ..
  <part class="G:Button" name="Submit"/>
   ..
</Structure>
```

Figure 6. Generated PIM1: Structure Part

```
<style>
  <property name="Background" value="WhiteColorTheme" part-name="Intervention_Request_Form"/>
  <property name="Font-Size" value="18" part-name="Intervention_N°"/>
  <property name="Font-Size" value="18" part-name="Sender"/>
  ..
  <property name="Font-Weight" value="Bold" part-name="Intervention_N°"/>
  <property name="Font-Weight" value="Bold" part-name="Sender"/>
  ..
</style>
```

Figure 7. Generated PIM2: Style Part

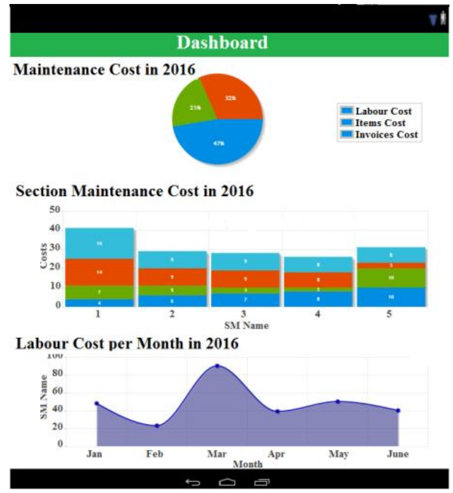

Figure 8. The generated Human-Computer interface

V. CONCLUSION AND FUTURE WORKS

In this paper, we presented a new ergonomic approach for designing context-aware Human-Computer interfaces. To model its ergonomic methodology, we used Artificial Intelligence techniques to design such systems. Finally, through the transformation rules, we have specified UI and produced UIML code that could be rendered in order to generate final interfaces for specific platforms. Ongoing and future works include the development of our approach with more case studies for ergonomic context-aware pervasive systems.

ACKNOWLEDGMENT

This project has been developed in the scope of a MOBIDOC doctoral thesis of the PASRI program sponsored by the European Union and administered by the ANPR. The authors would like to thank Mohammed Fouad Mansour Alzuhiry, who has also been active contributor during the implementation tasks and the company BMI (Tunisia) for financial and technical support.

REFERENCES

[1] J. Karat and C. M. Karat, “The evolution of user-centered focus in the human-computer interaction field,” IBM Syst. J. vol. 42(4), pp. 532-541, 2003.

[2] R. Yvonne, H. Sharp, and J. Preece, “Interaction Design: Beyond HumanComputer Interaction”, Wiley Publishing, 2011.

[3] C. Stephanidis, “Towards An information Society for All and Human Factors and Ergonomics,” Universal Access in HCI, vol. 3(1), Press, C.R.C, 2001.

[4] B. A. Myers, “Interface Software Technology,” chapter 72 of CRC Handbook of Computer Science and Engineering. Allen B. Tucker, editor in chief. Boca Raton, FL: CRC Press, Inc. pp. 1571-1595, 1997.

[5] M. Helander, Handbook of Human Computer Interaction, M. Helander (ed.), Elsevier Science Publishers B.V., North Holland, 1988.

 ISBN: 978-1-61208-498-5

[6] W.E. Gilmore, D.I. Gertman and H.S. Blackman, User-Computer Interface in Process Control, A Human Factors Engineering Handbook, Academic Press, 1989.

[7] M. Tendjaoui, C. Kolski and P. Millot, “An approach t owards the design of intelligent man-machine interfaces

[8] D.L. Scapin, P. Reynard and A. Pollier, “La conception ergonomique d'interfaces : problèmes de méthode, ” Rapport de recherche n° 957, INRIA, Décembre 1988. used in process control,” International Journal of Industrial Ergonomics, 8, 1991, pp. 345-361.

[9] J. Lazar, “Universal Usability: Designing Computer Interfaces for Diverse User Populations,” Chichester, UK: John Wiley and Sons, 2009.

[10] J. Nielsen, “Usability Engineering, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.

[11] S. Charfi, A. Trabelsi, H. Ezzedine, and C. Kolski, “Towards ergonomic guidelines integration within graphical interface controls for the evaluation of the interactive system Application to a transport network Information Support System,” In: Logistics (LOGISTIQUA), pp. 76-82, 2011.

[12] ISO, ISO/IEC 9126-1: Software engineering, Product quality, 2003, Part 1,2,3 and 4, (JIS X 0129).

[13] OMG, MDA Guide Version 1.0., http://omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf, 2003, [retrieved: October, 2015].

[14] J. Tarby, H. Ezzedine, and C. Kolski, “Prevision of evaluation by traces during the software design of interactive systems: two approaches compared,” In A. Seffah, J. Vanderdonckt, and M. Desmarais (Eds.), Human-Centered Software Engineering: Architectures and Models-Driven Integration, Springer HCI Series, pp. 257-276, 2009.

[15] A. Trabelsi and H. Ezzedine, “Evaluation of an Information Assistance System based on an agent-based architecture in transportation domain: first results,” International Journal of Computers, Communications and Control 8: 2, pp. 320-333, 2013.

[16] S. Charfi, H. Ezzedine, and C. Kolski, “RITA: a useR Interface evaluaTion frAmework,” Journal of Universal Computer Science, pp. 526-560, 2015.

[17] A. Beirekdar, J. Vanderdonckt, and M. Noirhomme-Fraiture, “A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code,” Computer-Aided Design of User Interfaces III, Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces, Valenciennes, France, pp. 337-348, 2002.

[18] C. Mariage, J. Vanderdonckt, A. Beirekdar, and M. Noirhomme, “DESTINE: Outil d'aide à l’évaluation de L'ergonomie des sites web,” Proceedings of the 16 Conference on Association Francophone d'Interaction Homme-Machine, ACM, pp. 117-124, 2004.

[19] C. Tran, H. Ezzedine, and C. Kolski, “Evaluation of Agent-based Interactive Systems: Proposal of an Electronic Informer Using Petri Nets,” Journal of Universal Computer Science (J.UCS), 14(19), pp. 3202-3216, 2008.

[20] D. L. Scapin and J. M. C. Bastien, “Ergonomic criteria for evaluating the ergonomic quality of interactive systems,” Behaviour & Information Technology, pp. 220-231, 1997.

[21] J. Nielsen and R. Budiu, “Mobile Usability”, New Riders Press, ISBN 0-321-88448-5, 2012

[22] User Interface Markup Language (UIML), Version 4.0. Oasis, http://docs.oasisopen.org/uiml/v4.0/cd01/uiml4.0- cd01.html, 2008, [retrieved: October, 2015].

[23] JBoss Drools, http://www.jboss.org/drools/, [retrieved: March, 2016]

[24] C. Forgy, “On the efficient implementation of production systems,” PhD Thésis, Carnegie-Mellon University, 1979.

[25] OMG, BPMN – Business Process Modeling Notation Specification version 1.0. OMG Available Specification, 2006.

[26] Acceleo, https://eclipse.org/acceleo/, [retrieved: March, 2016].

[27] ATLAS group LINA & INRIA, ATL, “Atlas Transformation Language. ATL User Manual,” version 0.7, 2006.

 ISBN: 978-1-61208-498-5