Applying Knowledge Management in UI Design Process

Pablo Ribeiro Suárez
Faculdades Integradas de Patos
Rua Floriano Peixoto, 223, Centro
CEP: 58.700-300
Patos – PB – Brasil
pablo@dsc.ufcg.edu.br

Bernardo Lula Júnior,
Marcelo Alves de Barros
Universidade Federal de Campina Grande
Av. Aprígio Veloso, 882, Bodocongó CEP: 58.109-970
Caixa Postal: 10.106
Campina Grande – PB – Brasil
barros@dsc.ufcg.edu.br

ABSTRACT
This work describes the use of a Knowledge Management (KM) strategy to analyze the User Interface (UI) design process. This analisys approach produced: (i) an ontology to the UI design process and (ii) a new transforming approach to produce an interaction description from task description. The ontology is based on the definition of task and interaction metamodels. The new transforming approach defines an automatic mechanism to convert a description in another one. It is based on these metamodels and on a scenic metaphor. This transformation preserves the structural and temporal decomposition of the tasks, makes its context explicit and maintains the coherence between the descriptions. This new approach redefines the appropriate time for ergonomic and/or project rules usage on UI design process.

Author Keywords
UI design, task-based design, Knowledge Management.

ACM Classification Keywords
User interfaces, ergonomics, theory and methods.

INTRODUCTION
The UI design has evolved from a purely empirical approach (from some years ago) to the current approach based on the task and on the ergonomic and/or project knowledge. The empirical approach is based on a quick prototype construction, making use of the available graphic tools. Those tools enable the easy generation of an evaluation prototype. The issue attached to this approach is: the usage of ergonomic knowledge, or any other involved in UI design, takes place only on the prototype evaluation stage. That means having frequently to re-design the whole project.

Opposing that practice, UI design has evolved to a task-based approach. In this approach, the ergonomic knowledge and the project experiences are used early since the conception stage. This is done through the ergonomic analysis of the work (task modeling) and the adoption of rules that represent the ergonomic knowledge and the project heuristics. These rules help the designer on the acquisition of an interaction description from a task description. This approach has been broadly spread by its application on many methodologies and/or interface conception frameworks, such as: ADEPT [17], ALACIE [9], ERGOSTART [14], MACIA [7], MEDITE [13], TRIDENT [3], and others. In these methodologies the starting-point of UI design is to generate a task description with the help of conceptual tools (computer-aided or not), such as: CTT [20], ETAG [24], GTA [25], HTA [1], MAD [21], TAKD [5], TKS [16], TAOS [20], etc. Thereafter, this task description is transformed into a conceptual description of interaction, according to a particular interaction model, such as: AIM/CIM [17], SSI [9], ICS [14], MIC [7] or EDITOR [19]. This transformation process is generally accomplished with the help of rules that stablish relations between task model attributes to interaction model attributes. These rules generally come from ergonomic recommendations and principles and/or interface project practices experienced by designers. Conception experiences supported by this transformation have been presented improvements comparing to the prior approach: the reduction of modifications pointed by an evaluation process and the production of a prototype already presenting properties of an interface which will be more suitable and adapted to the user's needs.

However, despite these improvements, the use of task-based approach and the rules application bring some problems in a real UI design environment: difficulties to formalize ergonomic and/or project knowledge, difficulties to implement these rules in CAD tools and difficulties to reuse rules in different contexts [2, 14, 26].

In this paper we describe the application of a KM strategy in UI design. This strategy's application has led to the proposition of a new approach for the acquisition of a conceptual description of interaction based on the task description making no use of ergonomic and/or project rules in the first instance. This approach, based on a scenic metaphor, defines a new model (something between the task model and the interaction model), and allows: (i) to preserve the structural and temporal decomposition of the task, (ii) to makes its context explicit and (iii) to maintain the coherence between the descriptions in sight of any occured modification in one of them.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

TAMODIA'04, Prague, Czeck Republic.
Copyright ©2004 ACM 1-59593-000-0/04/0011…$5.00

KNOWLEDGEMENT MANAGEMENT IN UI DESIGN

In this work an iteractive KM model [10] has been used and adapted [23] for the UI design due to its reach and applicability in the activities concerning software design. According to this model, the efficient estimation and exploration of knowledge involved in conception and development processes depend essentially on the definition of methods that allow effectively to classify, to represent, to integrate and to use this knowledge.

Classification Method of Knowledge Associated with UI Design Process

In the adopted KM strategy, the knowledge classification is the activity that involves the capture and identification of the diverse knowledge coexistent UI design process, taking into account their diverse natures (task, scenario, user, interaction, ergonomy, heuristics, etc.). Classification method is needed to identify and uniformize the terminology associated with this design context. Along the UI design conception process, uniformization practices have been experienced by: Welie [27], who defines a specific ontology to the UI design context, however, with a restricted terminology to the specific task representation; by Welie and Veer [28], whose research is based on methods and ontologies in interdisciplinary designs; by Furtado [8], who defines an ontology-based method to the UI design; and by Limbourg, Pribeanu and Vanderdonckt [18], who, trying to uniformize the terminology associated to the task description, have produced an unified task metamodel from many formalisms for the task description.

Each of these experiences produced an uniformization conditioned to the UI design practices principles. Furthermore, the terminologies are conditioned to the methodology scope, what makes them representative only to that methodology specific praticioners.

In this work, the adopted knowledge classification method is based on the accomplishment of four steps: (i) definition of the criteria of representativity of the considered methodologies of UI design; (ii) study of the UI design process defined for each representative methodologies; (iii) capture of the terminologies associated to the processes of conception defined by each methodology; and (iv) definition of a reference base-terminology for the integration of the captured terminologies.

The criteria of representativity defined for the selection of the methodologies present in the state of art in UI design was: a) task-based methodologies considered must have a complete vision of the UI design process; b) methodologies considered must use well-defined formalisms, models, tools and ergonomic knowledge, characterized by structured and documented processes and, c) methodologies considered must be recognized by publications in the state of art and by the design practicioners community. The considered methodologies respecting these criteria were: ADEPT [17], ALACIE [9], ERGOSTART [14], MACIA [7], MEDITE [13] and TRIDENT [3].

From capture of the terminologies associated to the processes of conception defined by the each methodologies, presented in [23], we obtained a base-terminology of reference to integrate these different terminologies. The base-terminology attained consists of the following terms: Presentation, Evaluation, Scenario, Client, Ergonomic Knowledge, Task Description, Dialogue, Sketch, Conceptual Specification of the Interaction, Interaction, Interface, Task Model, Architectural Model, Interaction Model, User profile, Interaction Objects, Designer, Prototype, Rules, Task and User.

Representation Method of Knowledge Associated with UI Design Process

The chosen knowledge representation method was based on the collaborative approach to construction of ontologies [12]. This approach has four distinct stages: (i) preparation for ontology construction, (ii) definition of a seed ontology, (iii) ontology construction and (iv) ontology validation.

In the preparation stage the evaluation criteria defined were: intelligibility, correctness, objectivity, usefulness and extensibility. The range defined includes the implicit knowledge (project heuristics) and the explicit knowledge (guides, patterns and recommendations) in UI design process. Ergonomics, Software Engeneering, UI design and evaluation processes were defined as content scope of the ontology.

As a seed ontology the base-terminology obtained in the classification method, described earlier, was used.

The ontology construction was performed in four steps:

1. Definition and identification of members of the envolved research teams as the ontology actors;
2. Definition of a form as information capture tool and in place visits integrated with email use as collaboration strategy.
3. Execution of interviews;
4. Analysis of the contributions and execution of feedback to the involved people (actors); and

This process was executed interactively, iteratively and incrementaly. So, the defined ontology covers all the UI Design Process cicle and is explicited trhought the metamodels observed in this cicle. Among several models found in UI design process the essential models are the task model and the interaction model. The proposed task and interaction metamodels were obtained from a generalization and uniformization of the terms present in the task and interactive models used in the several methodologies considered in this work. The ontology validation step will be presented in case study section.

To describe the task and the interaction metamodels the context-free grammar formalism was used as representation language, with the following notation: (i) the terms between “<” and “>” are variables (nonterminal symbols) of the grammar; (ii) the terms *in itallic* are metaterminals that are defined and valued in the specific formalism to be considered; and (iii) the terms between braces make a reference to a list of elements of a same nature.

Task metamodel

The proposed task metamodel is completely described in [23]. In this paper just a single extract from the grammar is presented for understanding the obtained ontology. In this description, the attributes IsScenario, IsScene and IsTake must be momentaneously ignored, for they'll be considered later, in the Script Metamodel definition.

<Task>::= *Identifier*
Name
Description
...
ParentTask
IsScenario
IsScene
IsTake
...
AdditionalComments

...

<Action>::= *Identifier*
Name
Description
...
ParentTask
IsScenario
IsScene
IsTake
...
AdditionalComments

...

Interaction metamodel

As in the task metamodel, the proposed interaction metamodel is also completely described in [23]. In this paper, only a grammar extract (essential for understanding the obtained ontology) is presented. The attributes ScenarioLink, SceneLink and TakeLink must be momentaneously ignored, for they'll be considered later, in the Script Metamodel definition.

<Interaction>::= <Presentation><Dialogue>

<Presentation>::= {<Space>}

<Space>::= *Identifier*
ScenarioLink
Description
<SpacePresentation>
{<Vision>}
...
AdditionalComments

...

<Vision>::= *Identifier*
SceneLink
Description
<VisionPresentation>
{<InteractionObject>}
...
AdditionalComments

...

<InteractionObject>::= *Identifier*
Description
TakeLink
<InteractionObjectPresentation>
...
AdditionalComments

...

The <Dialogue> component define the flow of interaction (sequences of messages) between the user and the application, and is described in the form of a grammar representing a diagram of states transition, where the states represent the elements of <Presentation> (<Space>, <Vision> and <InteractionObject>) and the arcs represent stimulus used in the communication process.

<Dialogue>::= {<State>}
{<Event>}
<InitialState>
{<FinalState>}
<TransitionFunction>

...

Integration Method of Knowledge Associated with UI Design Process

As started before, the approach for the obtainment of a conceptual specification of the interaction, based on the usage of ergonomic rules and/or design rules brings operational difficulties, as for the actors involved in the production of practical rules of transformation for a description into another, as for the interface designers, users of these rules.

In the adopted KM strategy, the integration of the knowledge is the activity that transforms task description in an interaction description, preserving the structural and temporal decomposition of the task and the context characterization of its execution. In order to find an operational alternative to the problem of these tranformation that will not reincide on the mentioned difficulties, we use the *divide and conquer principle*. According to this principle, an original problem is divided in two or more smaller problems with simpler solutions. In this work, this principle was combined with an integration method of the knowledge in the UI design process, to produces a method to transform a task description into an interaction description. This is made in two steps: (i) the definition of an intermediate metamodel between the task and interaction metamodels and (ii) the definition of mechanisms of transformation from the task model into the intermediate metamodel and from the intermediate metamodel into the interaction metamodel, preserving the structural and temporal decomposition of the task, exposing its execution scenario.

Intermediate metamodel of the task and interaction metamodels

To define an intermediate metamodel that will satisfy the mentioned restrictions, we proposed the use of a scenic metaphor [23]. This metaphor allows better characterization of the elements associated to the problem of transforming one description into another.

The word metaphor comes from the Greek *Metaphora*, which means “meaning transportation”. A metaphor is characterized by the description of a domain named target-domain, related to another one, called source-domain [15]. A set of inferences is associated with the source-domain, allowing one to think about that domain. The principle of application of a metaphor is the projection of a set of inferences of the source-domain over the target-domain. That induces transference of structural relations and thought mechanisms from the source-domain to the target-domain.

The partitioning of the problem of transformation from a task description in an interaction description in two sub-problems possesses a strong analogy with the problem of transposing a book to the movie or theatre format [11]. The movie or theater director does not make the movie or a drama (play) directly from the book. Instead, he/she uses an element of description called “script”. The script implements an adaptation from the book to the cinematographic or theatrical language.

A script is defined, among other acceptions, as a document enclosing the detailed descriptions of the scenarios, scenes, actions and dialogues of a movie [4, 6]. A scenario, by its turn, is the space of representation where the scenes take place as a part of the film, drama, etc. A scene is defined as a vision or picture of each of the situations or actions along a film or drama. Finally, a take is defined as an act involving actors and objects.

According to these definitions and terminology, one can describe a script as a sequence of scenes that shall be shot or played. Each scene happens in a scenario, which describes the elements (decoration and action elements) present during each scene, and accomplished by the many takes (simple actions) to be executed with the involved actors.

Having in mind the method of integration (transformation) of knowledge adopted, the development process of a movie or drama (scenic metaphor) and the application principle of a metaphor was proposed a metamodel intermediate between the task and the interaction metamodels, the script metamodel [23]. The script metamodel is presented, below, according to the same formalism employed in the representation of the previously described task and interaction metamodels.

<Script>::= {<Scenario>}

<Scenario>::= *Identifier*
Description
AddicionalComments
TaskLink
{<Agent>}
{<Tool>}
{<Object>}
{<Effect>}
[{<Scenario>}]
{<Scene>}

<Agent>::= *Identifier*
...
AddicionalComments
{Association (*Role*,*TakeLink*)}

<Tool>::= *Identifier*
...
AddicionalComments
{Association (*Utility*,*TakeLink*)}

<Object>::= *Identifier*
...
AddicionalComments

<Scene>::= *Identifier*
Description
AddicionalComments
TaskLink | *ActionLink*
ScenarioLink
{<Effect>}
{<Take>}

<Take>::= *Identifier*
Description
AddicionalComments
ActionLink
SceneLink
{<Effect>}
{Association (*Agent*,*Tool*,*Object*)}

<Effect>::= Association (*Event*,*TakeLink*) |
Association (*Event*,*SceneLink*) |
Association (*Event*,*ScenarioLink*)

Thus, the application of the divide and conquer principle and the adoption of a scenic metaphor to obtain the interaction description, has introduced, in this process, another metamodel (script metamodel) in a way to suit the source and target domains to the structure precognized by the principle of application of a metaphor [15].

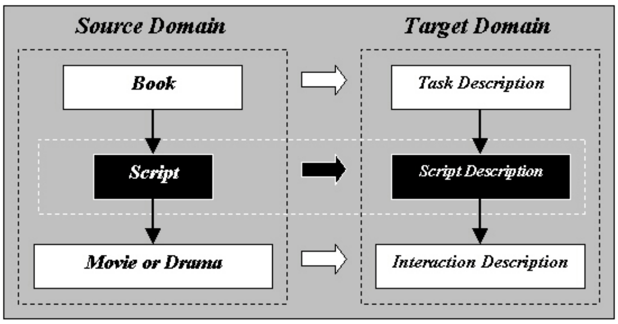

Figure 1. Process of transference of meaning used in the domain of the UI design.

The Figure 1 presents the process of meaning transfer employed in the UI design process. The tranformation problem of a task description in an interaction description may be partitioned in two subproblems: transform a task description in a script description and tranform the script description in an interaction description.

Mechanisms for transformation

The transformation of the task description in the interaction description, with the introduction of the script metamodel, must be done in two moments: the transformation of the task description in a script description and the transformation of the script description in an interaction description.

The transformation of the task description in a script description would be guided by the scenic metaphor, by the application of structural rules capable of associating elements present in the task metamodel (<Task> and <Action>) to elements present in the script metamodel (<Scenario>, <Scene> and <Take>). The rules applied to this transformation are as follows:

Rule	Rule Description
Rule Take01	*If Task is one <Action> (TM) defines one <Take> (SM) associate to Task*
Rule Scene01	*If Task is one <Task> (TM) whose decomposition presents at least one <Action> (TM), defines one <Scene> (SM) associated to Task*
Rule Scenario01	*If Task is one <Task> (TM) whose decomposition presents at least one <Task> (TM) that is associated with one <Scene> (SM), defines one <Scenario> (SM) associated to the Task*
Rule Scenario02	*If the Task is one <Task> (TM) that is associated with one <Scene> (SM) and at least one of its brothers is one <Task> (TM) associated to one <Scenario> (SM), associates with Task one <Scenario> (SM)*
Rule Scene02	*If Task is one <Action> (TM) that she is one son <Task> (TM) that is associated with one <Scenario> (SM), associates with Task one <Scene> (SM)*

Figure 2. Rules to associate elements presents in the task metamodel to elements of the script metamodel.

Induced by the scenic metaphor, there is a bi-reciprocical relation between the script and interaction metamodels [23]: a scenario is defined as the space of representation where scenes take place; a scene is defined as a vision of each of the situations that came out; and, by its turn, a take is defined as the representation of the manipulation of objects by the user. That way, this relation inducts an isomorphism between the script and interaction metamodels and is defined by structural rules that associate elements of the script metamodel (<Scenario>, <Scene> and <Take>) with elements of the interaction metamodel (<Space>, <Vision> and <InteractionObject>). Those rules, defined in the picture below, allow the tansformation of a script description into an interaction description.

Rule	Script metamodel concept	Interaction metamodel concept
Rule Space01	Scenario	Space
Rule Vision01	Scene	Vision
Rule InteractionObject01	Take	InteractionObject

Figure 3. Rules capable to associate elements presents in the script metamodel with elements of the interaction metamodel.

Structure Preservation and Context Characterization

The adopted method of knowledge integration defined above precognizes, not only the definition of an intermediate metamodel between the task and interaction metamodels and the mechanisms of transformation, but also the satisfaction of the restrictions stated there: the preservation of the structural and temporal decomposition of the task and the characterization of the context of its execution.

Regarding the preservation of the structural decomposition of the task, it can be observed that the structure of the script description (a tree), obtained by the application of the rules of tansformation of the task description into the script description, maintains the structure of the task description (task tree). That is an isomorphism.

The preservation of the temporal decomposition of the task is also achieved, for the temporal operators of the task are visible to the script metamodel throught the attributes TaskLink and ActionLink and to the interaction metamodel throught the attributes ScenarioLink, SceneLink and TakeLink. These links serve not only to ensure the preservation of the temporal decomposition of the task, but also as a mechanism to ensure the maintenance of the coherence among the involved metamodels caused by any modifications occurred in any of the metamodels. The Figure 4 below illustates such a mechanism:

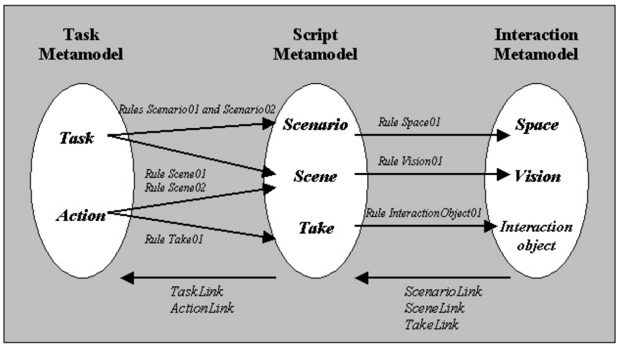

Figure 4. Mechanism for the maintenance of the coherence among the elements of the task, script and interaction metamodels.

Regarding the characterization of the task execution context, the script metamodel, by the definition of <Scenario>, allows the explicitation and the aglutination of all of the agents, tools and objects direltly involved with the execution of the specific task alluded to by the attribute TaskLink.

THE NEW APPROACH

The introduction of a new metamodel in the process of transformation of a task description into an interaction description has allowed the proposition of a new approach to the process that makes use of only a restricted set of strutural rules and demands no interpretation nor contextualization by the designer.

The process of obtainment of a conceptual specification of the interaction precognized by the new approach consists of the following steps: (i) task analysis, (ii) obtainance of the script description and (iii) obtainance of a partial conceptual specification of the interaction from the script desciption. From that partial conceptual specification, ergonomic and/or design rules can be used to define characteristics of the interaction that are not related with the structure nor with the task execution sequence.

This approach prescribes the production of a partial conceptual specification of the interaction that must be processed right away, with the support of ergonomic and design knowledge, to generate the complete interaction conceptual specification.

The partial interaction conceptual specification concerns the structure of the <Presentation> (<Space>, <Vision> and <InteractionObject>) and enclosures the necessary elements for the <Dialogue> production (<TransitionFunction>). This specification is obtained from the script description, which, by its turn, is obtained by the task description, making use of the set of structural rules.

The complete interaction conception specification (<Presentation> and <Dialogue>) can be obtained from the partial interaction conceptual specification with the help of (i) ergonomic and/or design rules to fulfill the attributes that define the interaction style, the appropriate interaction objects and their characteristics, such as location, size, color, etc., and (ii) other structural rules for the construction of the states transition diagram.

CASE STUDY

We applied the new approach in a real situation: the conception of an interactive site used as a tutorial to help on the teaching of Turing machines. In this tutorial, all information on the proposed subject must be made available in the most intuitive and pleasing way. Also, internet possibilities must be available for easy exploitation by the user.

Task Analisys

For task analisys we used the meta-model proposed in this work. Figure 5 ilustrates (using TAOS language) the hierachichal structure of the root-task Accessing web-tutorial (1), which has as subtasks: Study (1.1), Do exercices (1.2), See references (1.3), Visit related sites (1.4), Downloads (1.5) and Ask for help (1.6).

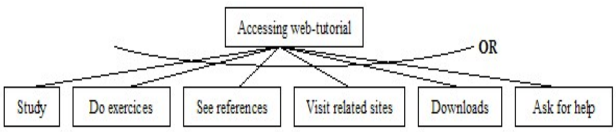

Figure 5. High level of task-tree for Accessing web-tutorial task.

Sub-task Study (1.1) is composed by sub-tasks Choose subject (1.1.1) and Study subject (1.1.2). Study subject (1.1.2) has the sub-tasks (actions) Choose specific topic (1.1.2.1) and Study specific topic (1.1.2.2). Sub-task Do exercices (1.2) is composed by sub-tasks Choose exercices (1.2.1) and Resolve exercices (1.2.2). Sub-task See references (1.3) is composed by sub-tasks Choose references (1.3.1) and Read references (1.3.2). Sub-task Visit related sites (1.4) is composed by sub-tasks Choose site (1.4.1) and Access site (1.4.2). Sub-task Downloads (1.5) is composed by sub-tasks Choose file (1.5.1) and Download file (1.5.2). Sub-task Ask for help (1.6) is composed by sub-tasks Ask for help on subject (1.6.1) and Ask for help on site (1.6.2). Sub-task Ask for help on subject (1.6.1) is composed by sub-tasks See FAQs (1.6.1.1) and Contact tutor (1.6.1.2). Sub-task Contact tutor (1.6.1.2) is composed by sub-tasks Choose contact (1.6.1.2.1) and Do contact (1.6.1.2.2). Sub-task Ask for help on site (1.6.2) is composed by sub-tasks See map (1.6.2.1) and See scheme (1.6.2.2). Sub-task See map (1.6.2.1) is composed by sub-tasks See graphic map (1.6.2.1.1) and See hypertextual map (1.6.2.1.2). We can follow this using the structure described in Figure 6.

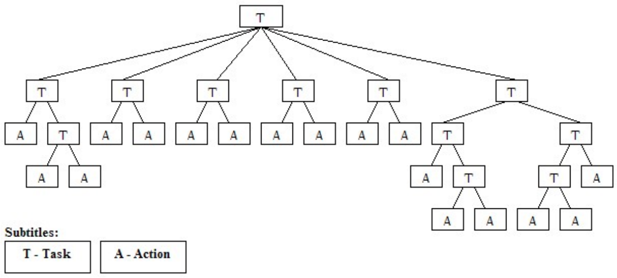

Figure 6. Complete task-tree for Accessing web-tutorial task.

From Task Description to Script Description

In this step the rules from Figure 2 were applied to transform elements of task description (tasks and actions) into elements of the script description (scenarios, scenes and takes). These rules transformed the task-tree into a script-tree.

The first rule (Take01) associates (or reclassifies) the Actions (leeves of task-tree) of task description to the Takes (leeves of script-tree) of script description.

The second rule (Scene01) associates (or reclassifies) the Tasks that have some Action in your decomposition to the Scenes of script description.

The third rule (Scenario01) associates (or reclassifies) the Tasks that have some task in your decomposition that was transformed into a Scene, to the Scenarios of script description.

The fourth rule (Scenario02) associates (or reclassifies) the Tasks that have some sister-task (task that have same value for *ParentTask* attribute) that was transformed into a Scenario, to the Scenarios of script description.

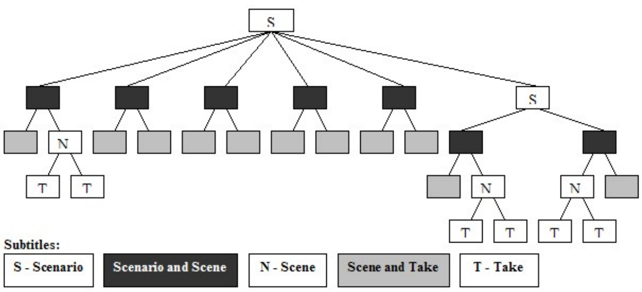

Figure 7. Complete script-tree for Accessing web-tutorial task.

The last rule (Scene02) associates (or reclassifies) the Actions that are daughter of a Task that was transformed into a Scenario, to the Scenes of script description. The Figure 7 below illustrates the script-tree.

From Script Description to Interaction Description

In this step, according to the rules from Figure 3, elements of script description (scenarios, scenes and takes) were transformed into elements of interaction description (spaces, visions and interaction objects). These rules transformed the script-tree into an interaction-tree as seen in Figure 8. This tree is isomorphous related to task-tree and script-tree presented in Figures 6 and 7.

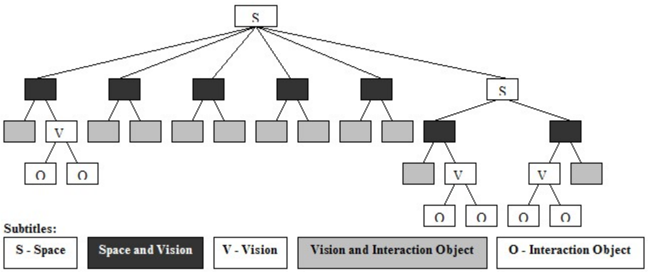

Figure 8. Complete interaction-tree for Accessing web-tutorial task.

From the nodes (spaces, visions and interaction objects) explicit in the interacton-tree and from the maintenance of coherence links (TaskLink, ActionLink, ScenarioLink, SceneLink and TakeLink in Figure 4), it is possible to construct the states-transistion diagram that defines the dialogue component of the interaction. From the partial conceptual specification, represented by interaction description, we used additionally ergonomic knowledege to generate a complete interaction specification that allowed the prototyping of the interfaces presented in Figure 9.

The validation in this case study was executed through the confrontation of the interface prototypes obtained through the useage of MEDITE [13] methodology and through the usage of the new approach. In this case study, the usage of the new appoach has managed to produce, with little effort by the designer and free from interpretation and contextualization errors, the same interface set produced by the use of MEDITE.

Figure 9. Prototype of Accessing web-tutorial and Study tasks.

CONCLUSION

The adoption of a KM strategy for the systematic study of UI design allowed a valorization and a more efficient exploration of the knowledge involved in the process. Main expressive results of this analysis, are: (i) metamodels for the essential knowledge concerned on transforming a task description into an interaction description; (ii) automatic mechanism to transform a task description into a partial conceptual description of the interaction, that preserves the task structural and temporal decomposition, makes its context explicit and allows the maintainance of the coherence among the involved descriptions in the process; (iii) formalization of the scenario notions through the definition of the script metamodel; and (iv) a new approach to UI design, using metamodels and the mechanisms of transformation of the task description into a partial description of the interation.

Besides these theoretical and practical results and the contribution for the insertion of KM practices in UI design, this work demonstrated the need and the opportunity to review the current pratices precognized by the UI design methodologies. A new element (script model) for UI design that allows a double vision on its usage (functional and didactic) was introduced. Case study also demonstrated no difficulties to formalize ergonomic and/or project knowledge, or to implement these rules in CAD tools or to reuse rules in different contexts.

REFERENCES

22. Annett, J., Duncan, K., “Task analysis and training design”. Occupational Psychology 41 (1967) 211-227.

1. Barbosa, S. D. J., Souza, C. S. de, Paula, M. G. de, Silveira, M. S., “Modelo de Interação como Ponte entre o Modelo de Tarefas e a Especificação da Interface”, IHC’2002.

2. Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Provot, I., Sacré, B., Vanderdonckt, J., "Towards a Systematic Building of Software Architectures: the TRIDENT methodological guide", in Proc. of 2nd Eurographics Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'95 (Toulouse, 7-9 juin 1995) 1995, pp. 262-278.

3. Bueno, S., “Mini-dicionário da língua portuguesa”, (Edição para o ensino fundamental) FTD, São Paulo, 2000.

4. Diaper, D., “Task Analysis for Knowledge Descriptions (TAKD): the method and examples” in DIAPER, D. (ed.): Task Analysis for Human-Computer Interaction, Ellis-Horwood (1990) 108-159.

5. Ferreira, A. B. de H., “Mini-dicionário”, 2ª Edição Revisada e ampliada, Editora Nova Fronteira, Rio de Janeiro, 1988.

6. Furtado, M. E. S., “Mise en oeuvre d’une méthode de conception d’interfaces adaptatives pour des systèmes de supervision à partir des spécification conceptuelles”, Thèse de doctorat. Université d’Aix Marseille III, France – 1997.

7. Furtado, E., Furtado, J. J. V., Silva, W. B., Rodrigues, D. W. T., Taddeo, L. da S., Limbourg, Q.,

Vanderdonckt, J., “An ontology-based method for universal design of user interfaces”, Proceedings of Workshop on Multiple User Interfaces over the Internet: Engineering and Applications Trends, A. Seffah, T. Radhakrishnan & G. Canals (éds.), Lille, 2001.

8. Gamboa, F. R., “Spécification et implémentation d'ALACIE: Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques”, Thèse de Doctorat, Paris XI, 1998.
9. Gartner, “Knowledge Management Scenario”, Conference Presentation, IT Symposium, 1999.
10. Gruen, D., “Beyond Scenarios: The Role of Storytelling in CSCW Design”. (Technical Report#: 00-02, Category - Work Practices), Submitted to CSCW - 2000.
11. Gruninger, M., LEE, J., “Ontology – Applications and Design” in Communications of the ACM, Vol. 45, Nº 2, (39-65) February 2002.
12. Guerrero, C. V. S., Lula, B. Jr., “Model-guided and task-based approach to UI design centered in a unified interaction and architectural model”, in Proc. of CADUI'2002 - 4th International Conference on Computer-Aided Design of User Interfaces, p. 107 - 119, Valenciennes, France, 2002.
13. Hammouche, H., “De la modélisation des tâches utilisateurs au prototype de l’interface homme-machine”, Thèse de Docteur, Université Paris VI, France, 1995.
14. Indurkhya, B., “Approximate transference: a computational theory of metaphors and analogies”, Cognitive Science, Vol. 11, 1987, p. 445-480.
15. Johnson, P., Johnson, H., “Knowledge Analysis of Task: Task Analysis and Specification for Human-Computer Systems” in Downton, A. (ed.): Enginnering the Human-Computer Interface. McGraw-Hill, Maidenhead (1989) 119-144.
16. Johnson, P., Wilson, S., Markopoulos, P., Pycock, J., ADEPT – Advanced Design Environment for Prototyping with Task models, INTERCHI’93 Conference Proceedings, Amsterdam: ACM, 1993.
17. Limbourg, Q., Pribeanu, C., Vanderdonckt, J., “Towards Uniformised Task Models in a Model Based Approach”, in Proc. of 8th International Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'2001, Ch. Johnson (eds.), Lecture Notes in Computer Science, Vol. 2220, Springer-Verlag, Berlin, 2001, pp. 164-182.
18. Lula, B. Jr., “Elaboration d’un Environnement de Génération Interactive d’Interfaces à Manipulation Directer pour le Language OBJLOG”, Thèse de Docteur, Universidade de Droit d’Economie et des Sciences d’Aix-Marseille III, Faculté des Sciences et Techniques de Saint-Jérôme, France, 1992.
19. Medeiros, H. e Rousselot F., “Un Outil D'Aide à la Modélisation de Concepts Dynamiques: Le Système TAME”; Journées Acquisition - Validation - Apprentissage, JAVA’95, 04/95, France, 1995.
20. Paterno, F., “Model based design and evaluation of interactive applications”, Spinger Verlag, Berlin, 1999.
21. Scapin, D., Pierret-Golbreich, C., “Towards a method for task description: MAD” in Berlinguet, L., Berthelette, D. (eds.): Proc. of Conf. Work with Display Units WWU’89, Elsevier Science Publishers, Amsterdam (1989) 27-34.
22. Suárez, P. R., “Gestão do Conhecimento no processo de concepção de IHC e uma nova abordagem para a obtenção de uma especificação conceitual da interação”, Dissertação de Mestrado - COPIN, UFCG, Brazil, 2004.
23. Tauber, M. J., “ETAG: Extended Task Action Grammar - a language for the description of the User’s Task Language” in: Diaper, D., Gilmore, D., Cockton, G., Shakel, B. (eds.): Proc. of 3rd IFIP TC 13 Conf. On Human-Computer Interaction Interact’90. Elsevier, Amsterdam (1990) 163-168.
24. van der Veer, G. C., van der Lenting, B. F., Bergevoet, B. A. J., “GTA: Groupware Task Análisis – Modeling Complexity”, Acta Psychologica 91 (1996) 297-322.
25. Vanderdonckt, J., Bodart, F., “Jusqu'au bout avec nos règles ergonomiques”, in Actes des Sixièmes Journées sur l'Ingénierie des Interfaces Homme-machine IHM'94, pp. 231-236.
26. Welie, M. V., “Task-based user interface design”, Dutch Graduate School for Information and Knowledge Systems, 2001.
27. Welie, M. V., Veer, G. V. D., “Ontologies and methods in interdisciplinary design”, In: Computer Science Education: Challenges for the New Millenium, pp. 143-158, 1999.