Task Flow Modeling in Electronic Business Environments

Cristea BOBOILĂ
Mathematics and Informatics Department, University of Craiova

In recent years, internet based commerce has developed as a new paradigm. Many factors such as "at home delivery", easy ordering, and usually lower prices contributed to the success of the e-commerce.
However, more recently, companies realized that one of the major factors in having a successful internet based business is the design of a user interface that is in concordance with the users' expectations, which includes both functionality and user-friendly features. The functionality feature of an e-business interface is one of the most important elements when discussing about a specific internet based business. In our paper, we present methods to model task flows for e-business interfaces. We strengthen our study with the design modeling of a practical scenario that may appear in an on-line commercial environment.
Keywords: *task flow, e-business, hierarchical model, user interface, human-computer interaction.*

1 Introduction

Along with the rapid evolution of applications from different domains, where the interaction between the user and the informatics system has a tremendous impact, the development of structured, advanced interfaces, also occurred. In this context, since the most of the users are not familiar with the way informatics applications work, a suggestive design is essential for graphical interfaces.
Our paper presents a practical approach of designing a commercial interface using a unified and solid notation system, based on concurrent task trees. We emphasize the fact that a coherent and comprehensive system is essential for developing the task flow that models a commercial interface. In this way, both the clients and the providers can nevertheless benefit. Furthermore, we describe a practical case of applying the modeling principles to approach a possible case from nowadays business situations.
We also want to point out that the developing of tools and techniques for model the task flow in an interface has been the object of current research in the human-computer interaction area. Important results have been registered in [1], [3], [6] that propose a hierarchical modeling based on concurrent task trees, and also in [2], [4] that focuses on multiple contexts of use.
The rest of the paper is structured as follows: Section 2 presents the design principles of task flow modeling. Section 3 introduces the most advanced system to this moment that can be used for task flow modeling. Section 4 describes the modeling and design of a commercial interface. Section 5 concludes the paper.

2. Design principles

In order to create an efficient interface that would be used to accomplish possible functions of the application, a clear and solid notation system for task flow modeling is needed. The notation system is required for a proper integration of the offered functionalities and interactive aspects [3]. The notations must be easy to understand and use, and must be able to structure large specifications employed in commercial applications.
Furthermore, it is essential that the notations have a precise semantics that would eliminate the possible ambiguities in communication [6]. In this context, the task flow modeling is very important for advanced interfaces, with the purpose to facilitate the work of those that develop interactive application, where the dialog human-computer is essential. In the case of a commercial application, both the user and the implementer should have a clear view of the task that can be carried out using the interface.

We consider that the designing of a task flow should follow some essential steps, which are described below:
1. First, a logical decomposition of task must occur. The result is a hierarchy of tasks. The tasks are assigned to different levels and nodes, depending on the abstraction level, starting with the most abstract task at the root and going to more practical and concrete tasks at the leaves.
2. After that, we identify the temporal relations between tasks situated on the same level in the hierarchy.
3. Next, we identify the objects associated to each task and also the actions based on which the tasks on the same level will communicate.
The steps described ensure a logical design of the task succession that can be accomplished with a well-defined user interface.

3. System description

In our view, the task flow modeling should allow the specification of task information and relevance for the user. In this context, we oriented our practical study to the ConcurTaskTrees (CTT) tool [5]. In comparison with previous approaches, such as Hierarchical Task Analysis, ConcurTaskTrees provides a more divers set of operations, each of them having a precise meaning.
The CTT system, based on well-defined symbols, has accomplishments in two essential directions that determined us to use it for our practical applications. On one side, it optimizes the description of functions carried out by the application from the point of view of the user, who will better understand how to use the application. On the other hand, it will help the designer to better and coherently describe the main tasks carried out by the application he develops. These two aspects are essential in a commercial tool, where we want the interests of both the clients and the providers to be well and clearly represented and easily carried out towards a successful fulfillment.
CTT is a notation system, created in support for task modeling design. It uses XML representation, and also a set of graphical symbols, defining the operators used to describe the relations between different types of tasks. The result of application description with CTT is a graphical representation of the functioning of the application. The graphical representation has a tree-like structure, whose nodes represent the tasks executed at a specific time. Between these nodes, there are different relations expressed with the corresponding operators in the CTT system.
One of the advantages of the CTT system is the possibility to extend it to represent tasks that are created by the user and are dependent of the context [4]. When one or more characteristics are changed, we have to reconfigure the interface to adapt to the new context of utilization. The CTT notation was proposed to model user-defined tasks, based on their relevance to express the user's intentions and the activities that they must carry out to get to the desired result.
Moreover, the designer of the task flow model should be able to manage various temporal relations, which would establish a temporal ordering between tasks, such as concurrency, interruptions, going from active to inactive state, iterations, and so on. The CTT tool provides these facilities. It is the most advanced flow modeling technique at this moment. In comparison with one of its predecessors, UAN, that also uses hierarchical modeling, CTT has the benefit of providing an easy to interpret model, which is a great advantage for users. Furthermore, the CTT system delivers a concise final representation of the flow, describing possible evolutions during a user session.

4. E-commerce application: Database administration

The CTT representation can be used to successfully accomplish the model that we outlined in Section 2. The tool contains an XML version for CTT, and also facilitates the specification of interaction with respect to the Abstract User Interface. The tool also provides the definition, editing and evaluation of models that can be constructed either from the perspective of a unique user, or from the perspective of multiple users. Vari-

ous output formats of the task flow designed can be obtained, such as xml, jpeg, html.
The following application models the interaction with a database. A database is an essential component in any commercial site, since it stores all the history of transactions in the system.

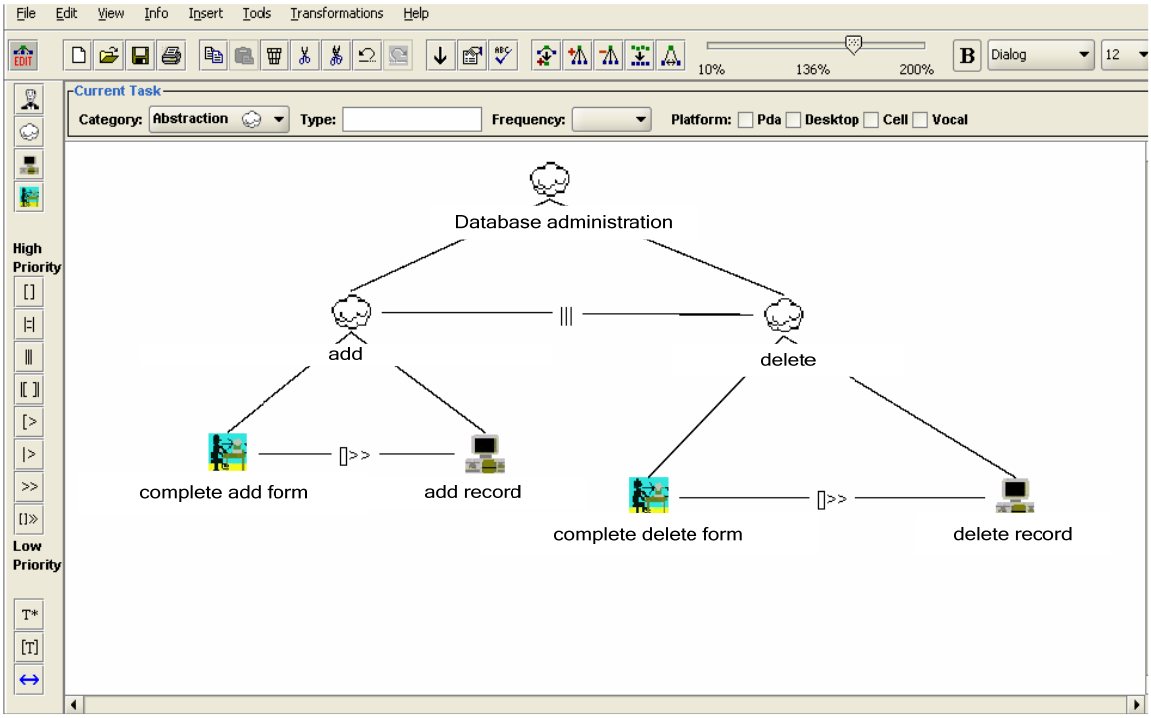

Fig.1. Database administration

The database administration has the associated root node, which is of abstract type. It has the two components shown in Figure 1, namely the adding and deleting of data, which are also modeled by abstract nodes, since they do not include a single concrete action in terms of user-computer interaction. As a general principle, the interaction with a database must allow adding and deleting of records. Between these two types of operations, a temporal order can not be established, because we can not firmly say that one precedes the other. Therefore, the link between these two operations is |||, representing the fact that two tasks can be performed concurrently.
The task that models adding data is of type *system-user interaction* task, that we have called "complete add form". The *system-user interaction* type has a particular icon, as shown in Figure 1. The "complete add form" task precedes and delivers data for the "add record" task, which is a different type of task, called *application* task. The symbolic notation for precedence accompanied by delivery of data is []>>. The *application* type is only associated with tasks that are performed entirely by the system. Similarly, deleting of data is conceptually modeled with a system-user interaction task and an application task, called "complete delete form" and "delete record" respectively. The "complete delete form" task precedes and delivers data to the "delete record" task.
Therefore, the flow modeled consists in the user filling forms for add or delete, followed by the corresponding action performed by the system. The relationship between the tasks is well expressed with the corresponding symbols in CTT notation. We have also used the tool for simulation, and observed the corresponding flow going through each task successively. The XML file associated is generated, as shown below:

```
<TaskModel NameTaskModelID="D:\IE\CTT\gctt.xml">
<Task Identifier="database administration" Category="abstraction"
Iterative="false" Optional="false" PartOfCooperation="false" Frequency="">
  <SubTask>
   <Task Identifier="add" Category="abstraction"
    Iterative="false" Optional="false" PartOfCooperation="false"
    Frequency="null">
     <TemporalOperator name="Interleaving"/>
     <Parent name="database administration "/>
     <SiblingRight name="delete"/>
     <SubTask>
       <Task Identifier="complete add form" Category="interaction"
       Iterative="false" Optional="false" PartOfCooperation="false"
       Frequency="null">
         <TemporalOperator name="SequentialEnablingInfo"/>
         <Parent name="add"/>
         <SiblingRight name="add record"/>
       </Task>
       <Task Identifier="add record" Category="application"
       Iterative="false" Optional="false" PartOfCooperation="false"
       Frequency="null">
         <Parent name="add"/>
         <SiblingLeft name="complete add form"/>
       </Task>
    </SubTask>
  </Task>
  <Task Identifier="delete" Category="abstraction"
  Iterative="false" Optional="false" PartOfCooperation="false"
  Frequency="null">
    <Parent name="database administration"/>
    <SiblingLeft name="add"/>
    <SubTask>
      <Task Identifier="complete delete form" Category="interaction"
      Iterative="false" Optional="false" PartOfCooperation="false"
      Frequency="null">
        <TemporalOperator name="SequentialEnablingInfo"/>
        <Parent name="delete"/>
        <SiblingRight name="delete record"/>
      </Task>
      <Task Identifier="delete record" Category="application"
      Iterative="false" Optional="false" PartOfCooperation="false"
      Frequency="null">
        <Parent name="delete"/>
        <SiblingLeft name="complete delete form"/>
      </Task>
    </SubTask>
  </Task>
 </SubTask>
</Task>
</TaskModel>
```

The XML output emphasizes the parent-child and sibling relationships. These relationships are marked by giving corresponding values to the elements: *Parent name, SiblingLeft,* SiblingRight, which refer to the parent and the two siblings on the left and right for the current node. Moreover, the temporal order is also recorded, in the *TemporalOperator* element.

5. Conclusions

The design of task flows is the base phase in the development of efficient e-business interfaces. We want the design process to be simple but coherent. Therefore, a suggestive notation of the steps performed by an application during its use is desired. This notation must describe the way in which the applica-

tion functions and the actions carried out by the user to get to the final result.

This paper studies fundamental aspects of flow modeling in e-business applications. Furthermore, we apply our principles to model a real application, represented by database administration on an on-line portal. In our opinion, the task flow must clearly express all the possible evolutions of an interactive system. In this context, we observe the applicability features of the ConcurrTaskTree system, which is based on a hierarchical model, and each evolution can be traced along branches, from the root to the leaves.

References:

[1] V.Penichet, F.Paternò, J.Gallud and M.Lozano, *Collaborative Social Structures and Task Modelling Integration*, Proceedings DSV-IS 2006, LNCS Springer Verlag, Dublin, 2006.

[2] F.Paterno, I.Santos, *Designing and Developing Multi-User, Multi-Device Web Interfaces*, Chapter 9, in Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006 Bucharest, Springer Verlag, 2006, pp. 111-122.

[3] D. Navarre, P.Palanque, F.Paternò, C. Santoro, R. Bastide, *A Tool Suite for Integrating Task and System Models through Scenarios*, Proceedings DSV-IS 2001, Lecture Notes Computer Science N.2220, pp.88-113, Glasgow, June 2001.

[4] N. Souchon, Q. Limbourg, J. Vanderdonckt, *Task Modelling in Multiple Contexts of Use*, Lecture Notes in Computer Science, Vol. 2545/2002, Interactive Systems. Design, Specification, and Verification: 9th International Workshop, DSV-IS 2002 pp59-73.

[5] G. Mori, F. Paternò, C. Santoro, *CTTE: Support for Developing and Analysing Task Models for Interactive System Design,* IEEE Transactions on Software Engineering, pp. 797-813, August 2002 (Vol. 28, No. 8), IEEE Press.

[6] F. Paternò, *Model-based Design and Evaluation of Interactive Applications,* Springer Verlag, November 1999, ISBN 1-85233-155-0.