Tailoring of ERP user interfaces using a model-based approach

Renate Kristiansen, renatek@idi.ntnu.no

Role-based access and portal solutions are considered the answer to the severe usability problems reported to torment users of ERP systems. An important issue related to role-based access is the combination of roles a given user have. A user can be represented in a Persona description containing a list of roles assigned to that person. But in the "real world" the composition of roles to people will differ among companies – hence flexibility in creating a user interface for different combinations of roles is important. We will suggest a systematic way to define what needs to be included in the user interface for one particular user based on her participation in the overall workflow using a combination of workflow models and models typically developed in the field of model-based user interface design.

Workflow model:

Workflow models focuses on how work is done to accomplish some organizational goals. A workflow partially order activities performed by humans or other actors in the enterprise in order to archive some predefined goals [Salimifard & Wright 2001]. Important workflow characteristics are tasks/activities that are performed by role-playing persons using supporting tools that gives access to various shared information resources [Marshak 1997] [Carlsen 1997].

Model-based user interface design (MBUID)

A task is an activity performed by people to accomplish a certain goal [Limbourg, 2001]. A high-level task can be decomposed into smaller task corresponding to lower level goals resulting in a tree structure. Task models are considered one of the viewpoints in the model-based community [Paternò, 2004]. Viewpoints are related to abstraction level (how much detail is considered?) and the focus (is the focus on the tasks or on the UI?) of the model. Usually the community works with models in four different viewpoints:

1. *Task and object model* represent the highest level of abstraction and their focus is on user's goals, tasks and the objects that are manipulated (the object model is often referred to as a domain model).
2. The second layer is the *abstract user interface* describing the structure and behavior of the user interface. Here the focus shifts to how the user interface can support the tasks specified in the task model, and connects tasks to objects, or parts of objects
3. The third level involves building *concrete user interface* specification defining the platform dependent look and feel of the interface.
4. Finally, the *final user interface* is the running interface implemented on a specific software environment.

Model-based user interface design (MBUID) processes often start with a task model that is evolved in an incremental process to the final user interface [Cuppens2006]. In each of the transformation phases the designer has the possibility to manually change the generated artifact, and the modification is preserved when regenerating the UI.

Relating workflow models with MBUID

The concept of task is very similar to that of process (in a workflow); the difference is mainly that of scope and focus. Processes typically relate directly to organizational goals, while tasks focus on the goal (and actions) of individual users playing certain roles in the organization. Hence, a task model may be seen as a refinement of a process model in the context of a specific user role [Trætteberg, 1999]. Figure 1 illustrates how a low-level workflow task can be the highest node in a task tree structure. The workflow model gives the context for the task tree decomposition and preserves the important workflow perspective in companies where colleagues cooperate to achieve a business objective.

The low-level task from the task model can be used as a specification for the abstract user interface. For example the Find PO (Purchase Order) task requires an AIO that allows the user to select one item from a set of items. The abstract user interface specification can be transformed to a concrete user interface model taking the platform into consideration. The final user interface is a result of interpreting or "running" the concrete user interface model.

In this paper the approach is presented as a linear process moving from the workflow diagram towards the final user interface using several models along the way. However, this is presented this way for explanatory reasons. For UI designers, it is also relevant to move from a concrete UI design towards the abstract design.

References

Carlsen, S. Conceptual Modeling and Composition of Flexible Workflow Models, PhD thesis, Norwegian University of Science and Technology. 1997

Cuppens, E., Raymaekers, C. & Coninx, K., A model-based design process for interactive virtual environments, Interactive Systems, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 225–236, 2006

Limbourg, Q. Pribeanu, C. Vanderdonckt, J., Towards Uniformed Task Models in a Model-Based Approach, LECTURE NOTES IN COMPUTER SCIENCE, 2001

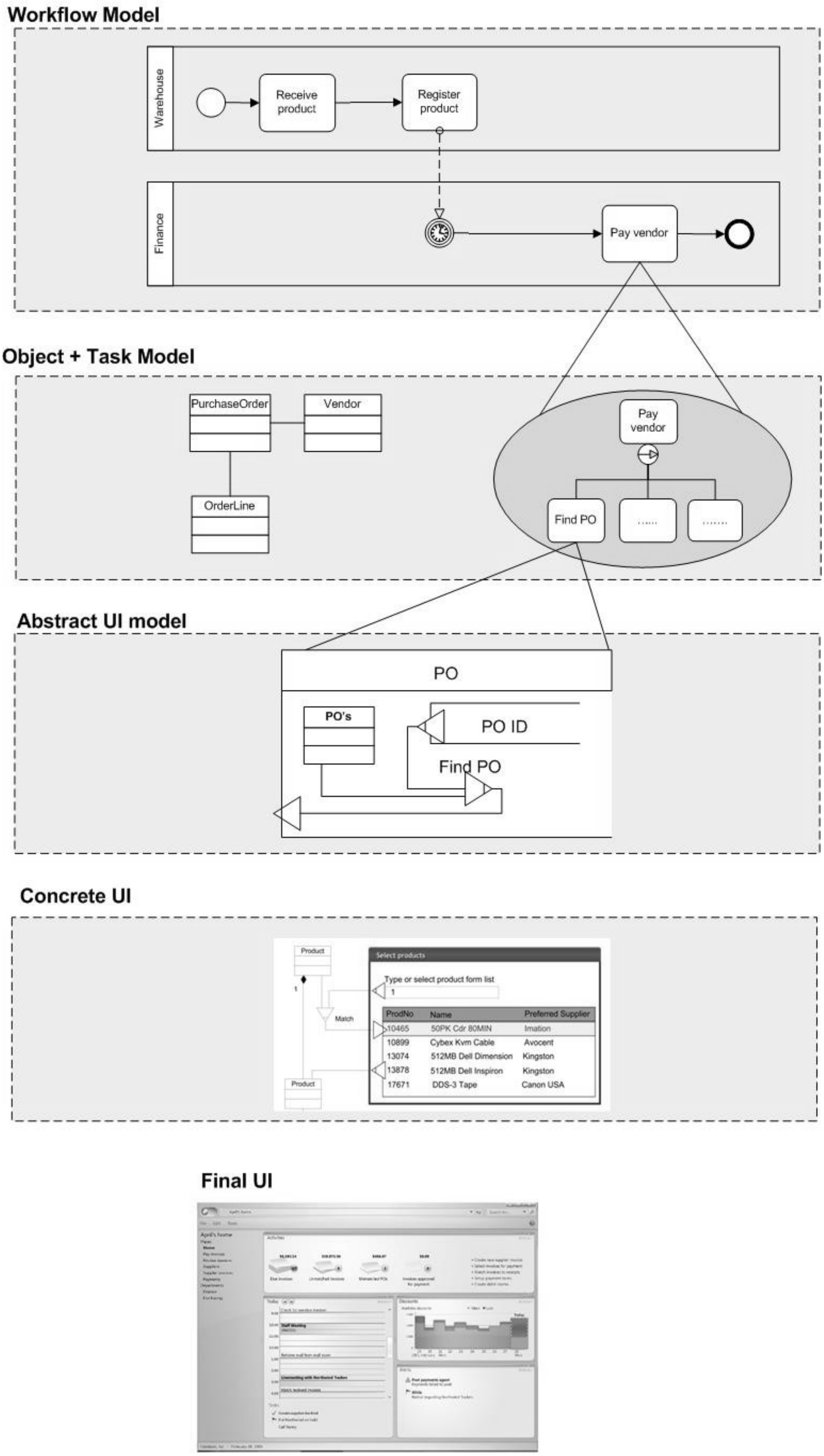
Marshak (1997). Workflow: Applying automation to group processes, in D. Coleman (ed.), Groupware - Collaborative Strategies for Corporate LANs and Intranets,

Salimifard, K. & Wright, M. (2001). Petri net-based modelling of workflow systems: An overview, European Journal of Operational Research, Vol. 134, pp. 664–676.

Paternò, F., Model-based tools for pervasive usability, Interacting with Computers
pp. 1–25, 2004.

Pribeanu, C. & Vanderdonckt, J. (2002). A methodological approach to task-based
design of user interfaces, Studies in Informatics and Control, Vol. 11, pp. 145–158.

Trætteberg, H. (1999). Modeling work: Workflow and task modeling, CADUI'99.



ProdNo	Name	Preferred Supplier
10465	50PK Cdr 80MIN	Imation
10899	Cybex Kvm Cable	Avocent
13074	512MB Dell Dimension	Kingston
13878	512MB Dell Inspiron	Kingston
17671	DDS-3 Tape	Canon USA

Figure 1: The relationship between different models